

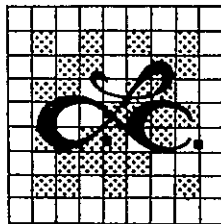
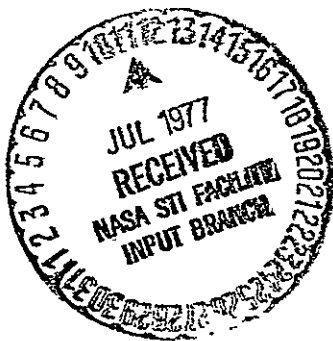
CR 151439

(NASA-CR-151439) HARDWARE SIMULATION OF
KU-BAND SPACECRAFT RECEIVER AND BIT
SYNCHRONIZER, PHASE 2, VOLUME 1 (LinCom
Corp., Pasadena, Calif.) 172 p.
HC A08/MF A01

N77-26348

Unclas

CSCL 17B G3/32 . 35434



LinCom Corporation

P.O. Box 2793D, Pasadena, Calif. 91105

HARDWARE SIMULATION OF KU-BAND SPACECRAFT
RECEIVER AND BIT SYNCHRONIZER
(PHASE II) - VOLUME I OF TWO VOLUMES

Prepared for

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

Technical Monitor: Jack Seyl

Prepared Under Contract No. NAS 9-14636

By

William C. Lindsey
Barry R. Eisenberg

LINCOM CORPORATION
P.O. Box 2793D
Pasadena, California 91105

March, 1977

TR No. 03-75-06-5

TABLE OF CONTENTS

	Page
SUMMARY	1
SECTION 1. INTRODUCTION	3
1.0 Discussion	3
1.1 Phase I Effort	3
1.2 Phase II Effort.	4
SECTION 2. PN SUBSYSTEM - LINCOM DESIGN	6
2.0 Functional Description	6
2.1 PN Acquisition Hardware and Logic	6
2.1.1 PN Synchronization Algorithms	9
2.1.2 Noise Estimate Algorithms	12
2.1.3 Acquisition Performance	12
2.2 Delay Lock Loop Hardware	16
2.2.1 Loop Behavior	18
2.2.2 Bandwidth Reduction	21
2.3 Loops Impact on Acquisition Time	23
2.3.1 T_{acq} Significantly Reduced in Positive Doppler	28
2.3.2 Recursive Filter Noise Estimate Effect on T_{acq}	32
2.4 PN Subsystem Integrated Performance	34

TABLE OF CONTENTS (Cont'd)

	Page
SECTION 3. PN SUBSYSTEM - NASA DESIGN	36
3.0 Introduction	36
3.1 Sync Algorithms	36
3.2 Software	36
3.2.1 Program Generating PN Acquisition Output Samples	39
3.2.2 PN Despreader Simulation Program	39
3.2.2.1 Input Parameters	40
3.2.2.2 Output Parameters	44
3.2.2.3 Options	44
3.3 Performance	44
3.3.1 ESTL Data	45
3.3.2 Performance Comparison	49
3.3.3 LinCom vs ESTL Algorithms - No Doppler	49
3.3.4 LinCom vs ESTL Algorithm - Doppler	53
SECTION 4. SYMBOL SYNCHRONIZER SUBSYSTEM	57
4.0 Introduction	57
4.1 Symbol Synchronization Techniques and Tradeoffs	57
4.1.1 Harmonic Generating-Tracking Symbol Synchronizers	59
4.1.2 Symbol Synchronizer Based Upon MAP Estimation Theory	61

TABLE OF CONTENTS (Cont'd)

	Page
4.1.3 Symbol Synchronizers Motivated by Optimum Design of the Phase Detector	63
4.1.4 Digital-Data Transition Tracking Symbol Synchronizers	65
4.1.5 Performance Comparison for Various Symbol Synchronizers	66
4.1.5.1 Tracking Mode/RMS Symbol Sync Jitter	66
4.1.6 Acquisition Performance Comparisons for the DTTL and Absolute Value Type Symbol Synchronizers	67
4.1.7 Bit Error Probability Degradation Due to Symbol Sync Jitter	69
4.1.8 Summary of Selected Symbol Synchronizer Configurations	69
4.2 Symbol Synchronizer Operation	71
4.3 Synchronizer Math Model and Loop Analysis	76
4.3.1 Phase Detector Characteristics for Sync Acquisition and Tracking	82
4.3.2 Variations in the Slope of the Phase Detector Characteristic for Acquisition and Tracking	83
4.3.3 Design Equations for Linearized Second-Order Loop	87
4.3.4 Symbol Sync Jitter Performance	90
4.4 Simulation Verification	90

TABLE OF CONTENTS (Cont'd)

	Page
4.4.1 Phase Detector Characteristic	90
4.4.2 Transient Response	93
4.4.3 Performance in Noise-Jitter	96
4.4.4 Performance in Noise - Error Rate	96
4.5 Software	96
SECTION 5. SYSTEM INTEGRATION	101
5.0 System Test Facility	101
5.1 Software Integration	101
5.1.1 Initialization	103
5.1.2 Acquisition and Tracking	106
5.1.3 Input Parameters	109
5.1.4 Output Parameters	113
5.1.5 Test Data	113
5.1.6 Options	117
5.2 Subroutines DLL, PNMON, RFSS, and DATGEN	118
5.2.1 Subroutines DLL and PNMON	118
5.2.2 Subroutine RFSS	121
5.2.3 Subroutine DATGEN	121
SECTION 6. SYSTEM STUDY	124
6.0 Introduction	124
6.1 System Baseline Performance	124

TABLE OF CONTENTS (Cont'd)

	Page
6.2 Measurement Accuracy	130
6.3 System Reconfiguration	130
6.3.1 Costas Arm Filter Unbalanced	134
6.3.2 Costas Arm Filters Removed	134
6.3.3 IF Filter Bandwidth Variations	134
6.3.4 Limiting	139
6.3.5 Channel Equalization	139
6.4 System Response to Signal Transients	143
6.4.1 Signal Fades	145
6.4.2 Carrier Phase Transients	147
6.5 PN/Carrier Sweep Interaction	147
6.6 PN Acquisition-Large Code Doppler Offsets	153
6.6.1 Doppler Limitation due to Search Strategies	156
6.6.2 Doppler Limitations due to Loop Behavior	158
6.6.3 Discussion	161
SECTION 7. CONCLUSIONS	163

SUMMARY

A hardware simulation of an automatically acquiring spacecraft receiver was developed for NASA/JSC during a previous study effort (Phase I). The information presented in this report represents work associated with a follow-on effort (Phase II). The Phase II objectives were:

1. Further investigate the PN Subsystem acquisition behavior.
2. Support NASA/JSC personnel in their use of the hardware simulation.
3. Construct a Symbol Synchronizer Subsystem which closely emulates the expected hardware.
4. Integrate this Symbol Synchronizer Subsystem into the composite simulation of the receiver.
5. Perform a series of system studies to evaluate overall performance of the receiver when subjected to anomalies such as signal fades and detect potential problems associated with PN/Carrier sweep interactions.

These objectives have been met with the following conclusions reached:

1. At a C/N_0 of 48.3 dB-Hz the mean acquisition time is expected to be approximately 50 seconds and relatively insensitive to doppler in this deep noise.
2. In the support of NASA/JSC personnel, a simulation was developed for their use which emulates the Harris hardware despreader. Preliminary comparison of the measured performance of the actual hardware with that predicted by the simulation indicates good agreement exists. It is expected that excellent agreement will be found once the software package is exercised by NASA in direct support of testing in a carefully controlled and monitored environment.

3. At the design point loop bandwidth of 30 Hz there will be little measurable degradation in performance associated with the Symbol Synchronizer Subsystem.
4. With the Symbol Synchronizer integrated with the remaining subsystems, its contribution to the overall degradation can be severe. Loss being a function of IF filter bandwidth and sampling rate associated with the digitally implemented synchronizer.
5. A series of experiments were performed to evaluate the systems behavior under signal transient conditions and the effects of sweep interactions on performance. No serious problems were uncovered, behavior being as expected. Additional studies performed to evaluate the effects of limiting, removing and unbalancing filters, bandwidth variations and channel equalization on performance proved to be more interesting.

SECTION 1

INTRODUCTION

1.0 Discussion

The Space Shuttle will receive two voice signals plus commands from the Earth over an S-band link which traverses the Tracking and Data Relay Satellite. This link is highly power limited and is also subject to international agreements limiting power density impingement on Earth, so the signal is spread by a PN code. The combination of these two actions results in a system that operates in a very low signal-to-noise environment and that is highly sensitive to degradations.

Relationships among key receiver system parameters, such as acquisition time, tracking resolution, and jitter are mathematically nontractable. In view of these analytical difficulties, and the system sensitivity to degradations, NASA/Johnson Space Center initiated developments of a time domain hardware simulation of anticipated receiver hardware to investigate the degree to which degradations will occur. The simulation was developed in two phases.

1.1 Phase I Effort

The receiver configuration was selected for simulation based on a series of system architecture and trade-off studies performed during the Phase I effort. This candidate receiver, capable of automatically acquiring and then tracking the spread waveform associated with a wideband PN modulated carrier was also simulated during the Phase I effort. Additional features of the design are automatic acquisition and tracking of the transmitted carrier and "idealized" coherent detection of the despread Manchester coded symbols. The

simulation and results obtained during the first phase effort are documented in Volumes I through III of the Phase I report.

1.2 Phase II Effort

The character and results of the Phase II effort are documented in this report. A review of the Phase I PN Acquisition study will be presented in Section 2. New material which discusses PN acquisition behavior in the presence of positive code doppler and bandwidth switching transients has also been included in this section.

The PN subsystem discussed in section 2 is based on a LinCom design. Harris, Inc. designed and constructed the actual hardware despreader which is identical to that simulated except for a series of sync algorithms. Section 3 describes a stand-alone PN despreader simulation software package, incorporating the algorithms employed by Harris. It is intended for use by NASA in supporting evaluation and testing of the actual hardware.

As previously stated an "idealized" symbol synchronizer, not capable of operating in deep noise, was included as part of the Carrier Recovery Subsystem during the Phase I effort. Its purpose was to evaluate the effects that carrier recovery hardware and PN despreader equipment have on symbol error rate performance. A Symbol Synchronizer Subsystem capable of tracking the Manchester encoded symbols in deep noise has been developed to replace the previous temporary design. Both the performance and simulation software for this subsystem are described in Section 4.

A discussion of the integration of the Modulator and Channel Subsystem, the RF front end and PN Subsystem, the Carrier Recovery Subsystem, and the Symbol Synchronizer Subsystem into the composite spread spectrum receiver and test facility will be presented, from a software standpoint, in Section 5.

The results and description of experiments performed to evaluate overall behavior of the integrated system are documented in Section 6. One series of experiments evaluates the symbol error rate performance as a function of filters and loop interactions and establishes the basic baseline performance. Another group of experiments evaluates the effects of limiting, removing and unbalancing filters, bandwidth variations, and channel equalization on overall performance. A third series of experiments evaluates system behavior in the presence of signal transients. A fourth group of experiments evaluates the effects of PN/Costas loop sweep interactions on system behavior. The final group of experiments investigates PN acquisition behavior in the presence of excessive amounts of code doppler.

Section 7 summarizes the total Phase II system engineering and programming effort. A series of appendices containing further, more detailed, software documentation are to be found in Volume II.

SECTION 2

PN SUBSYSTEM - LINCOM DESIGN

2.0 Functional Description

The RF front end and PN Subsystem are shown functionally in Figure 2.1. The incoming spread signal plus noise is passed through a variable gain amplifier and RF filter. The power level at the output of the filter is held constant by continuously adjusting the amplifier gain via a noncoherent AGC. The filter output is injected into a PN despreader which correlates a locally generated replica of the PN waveform $PN(t)$ with the incoming signal and provides to a microprocessor a voltage level which is a measure of the degree of correlation of the two waveforms. The microprocessor, in turn, steps the local PN generators code phase until correlation (PN alignment) is detected. A delay locked loop is then used to further adjust (fine tune) the local code phase so as to maximize the correlated signal seen by the Costas loop and data detector subsystem.

2.1 PN Acquisition Hardware and Logic

The PN acquisition hardware and logic are shown in Figure 2.2. The input to this module is the correlated signal plus noise waveform at the 2nd IF frequency of 80 MHz. This signal is filtered by a relatively narrowband IF filter, envelope detected and integrated over an interval T . At the sampling times the integrated output is A/D converted and transferred to the microprocessor. The integrator is then quenched. In the microprocessor the samples are compared to a threshold times an estimate of the noise power to determine the absence/presence of PN sync. The comparison algorithms as well as the algorithms used in obtaining the noise estimate are programmed into the microprocessor.

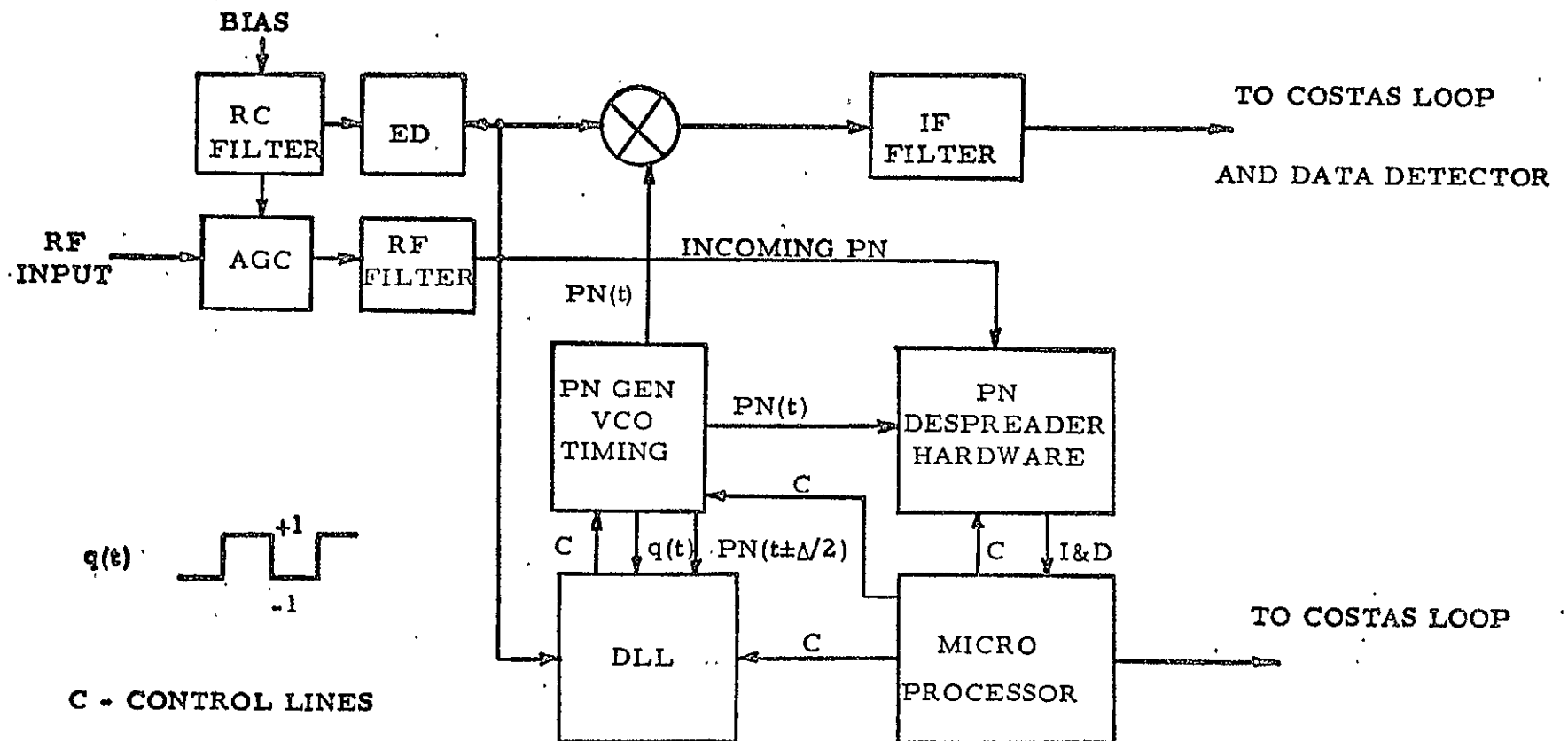


FIGURE 2.1. RF FRONT END AND PN SUBSYSTEM

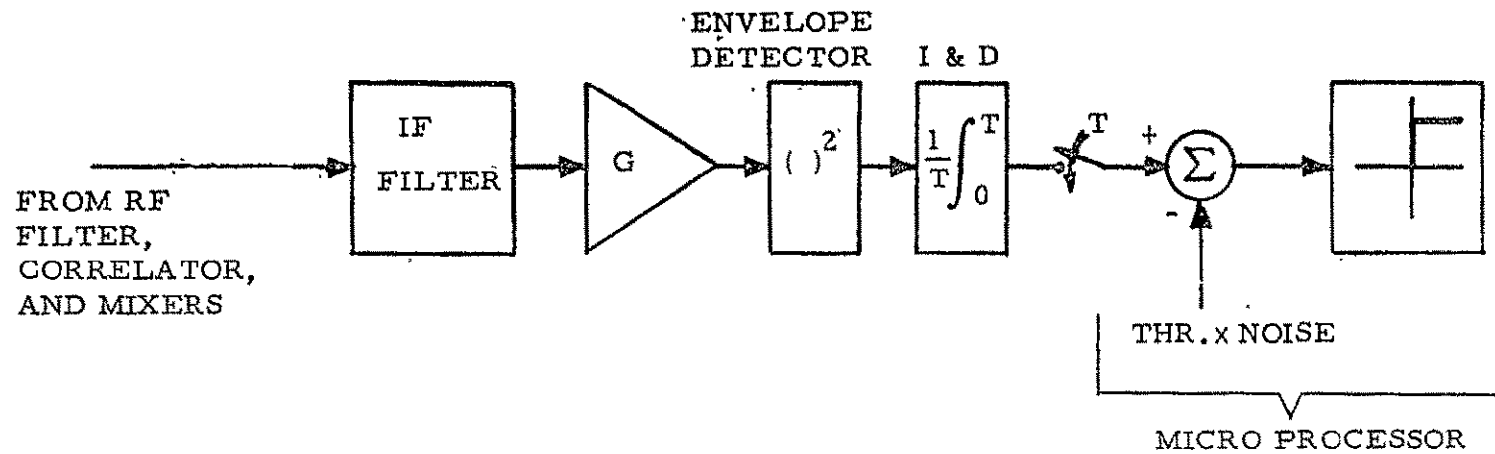


FIGURE 2.2. PN ACQUISITION HARDWARE AND LOGIC.

2.1.1 PN Synchronization Algorithms

The sync algorithms implemented in the LinCom design are presented in flow chart form in Figures 2.3 and 2.4. In the acquisition mode, the noncoherent output of the PN hardware is integrated over a duration T_1 . This integrated output is then compared to a threshold TH 1 times the estimate of the noise power. If the threshold is not exceeded then the noise estimate is updated, the delay locked loop is reset, the PN code moved 1/2 chip, and the search continued. If the threshold is exceeded, the output is integrated over a second time interval T_1 and another sync decision is made. Failure to obtain two hits in a row causes the code search to continue, with the noise estimate updated and the delay locked loop reset. Successfully obtaining two hits in a row initiates a sync verify operation with the threshold changed to TH 2 and the integrate time increased to T_2 . If the integrated output does not exceed the new threshold then failure to verify sync causes the resetting of the delay locked loop and the code chip search to continue. Once sync has been detected and verified, the Costas loop sweep starts and the tracking mode is entered.

In the tracking mode, a third threshold value, TH 3 is used (very low false alarm rate) with the integration time maintained at its present value of T_2 . The delay locked loop bandwidth is gradually narrowed. The system remains in the tracking mode as long as sync is found at least once in 20 trials. This provides a rather stiff fly wheel action which is required because of the rather large acquisition times inherent in operating at C/N_0 values less than 50 dB. Rapid fades and the bandwidth reduction process will not cause unwarranted loss of sync.

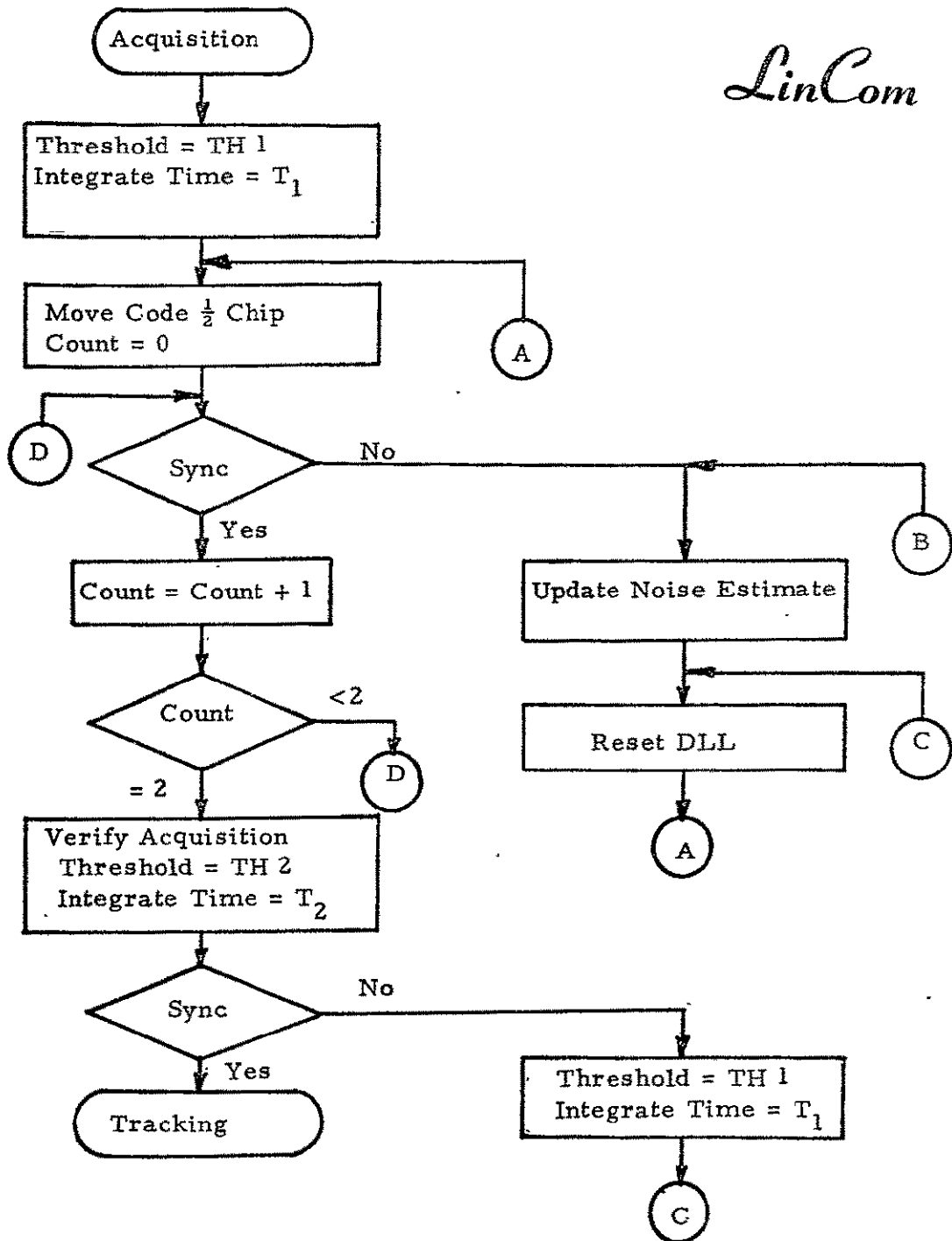


FIGURE 2.3. PN ACQUISITION AND VERIFY ALGORITHM.

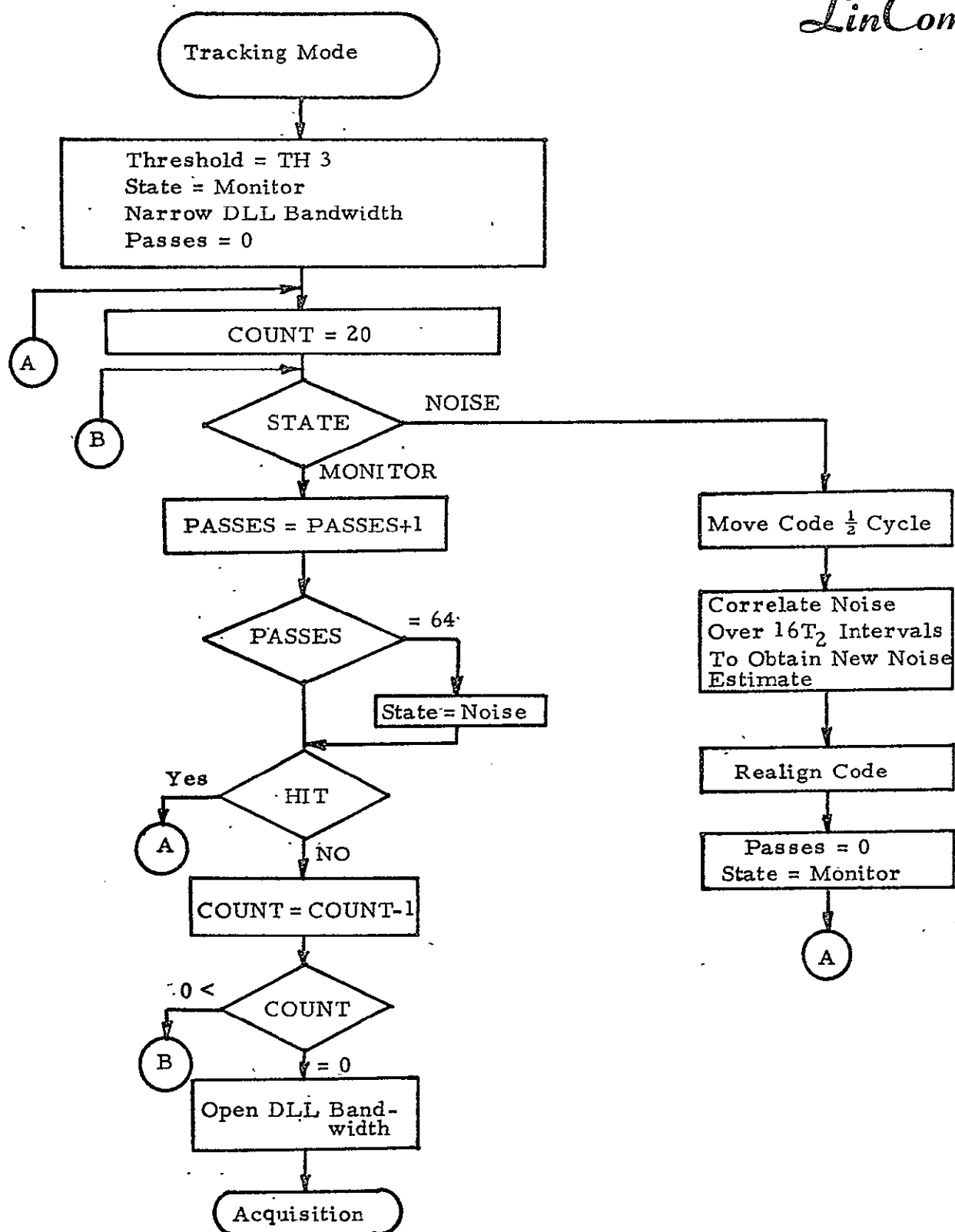


FIGURE 2.4. PN TRACKING ALGORITHM.

2.1.2 Noise Estimate Algorithms

Obtaining a good solid noise power measurement is critical because of the low C/N_0 values in which the system must operate. Common hardware is used to estimate both the signal plus noise power and the noise power. The algorithms to accomplish this are shown in Figures 2.5 and 2.6. As can be seen from Figure 2.5 sync will never be found on the first decision in acquisition. On the 2nd through 16th pass, sync is possible, with the noise estimate being averaged over successive samples. After the 16th sync decision the noise estimate is no longer updated by averaging but by a recursive filter technique which weights the most recent samples more heavily $[n_e(k) = \sum_{i=0}^{\infty} (\frac{1}{16})(\frac{15}{16})^i s_i(k-i)]$. In acquisition each time a sync indication occurs the value of s_i is saved. If the sync indication is not verified and the system does not enter the tracking mode then these saved values are included in updating the noise estimate. Once PN sync is found the algorithm used to obtain the noise estimate is shown in Figure 2.6. Every 64 sampling times, the PN code phase is shifted a half code cycle and 16 consecutive samples are averaged to obtain a new noise measurement.

2.1.3 Acquisition Performance

Curves of theoretical average acquisition time, T_{acq} , versus C/N_0 both in the absence and presence of doppler are plotted in Figure 2.7. These curves are based on the closed form expression

$$T_{acq} = (\frac{2-p}{2p})T_e$$

where T_e is the effective time to completely cycle all the code phase positions once in the absence of signal and is given by

$$T_e = \frac{2 \times 2047 T_D}{1 - 2\Delta f T_D}$$

LinCom

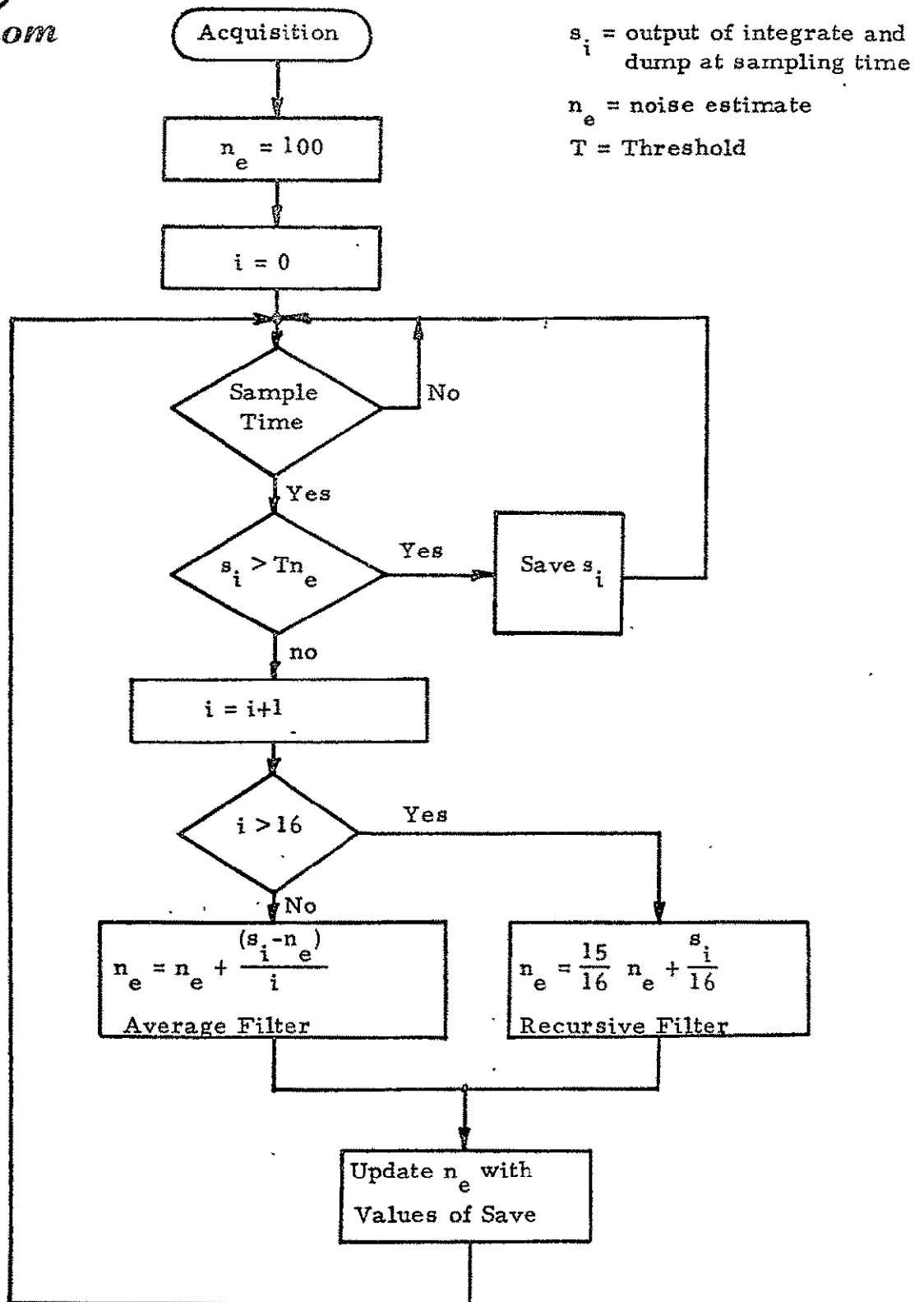


FIGURE 2.5. ACQUISITION NOISE ESTIMATE ALGORITHM.

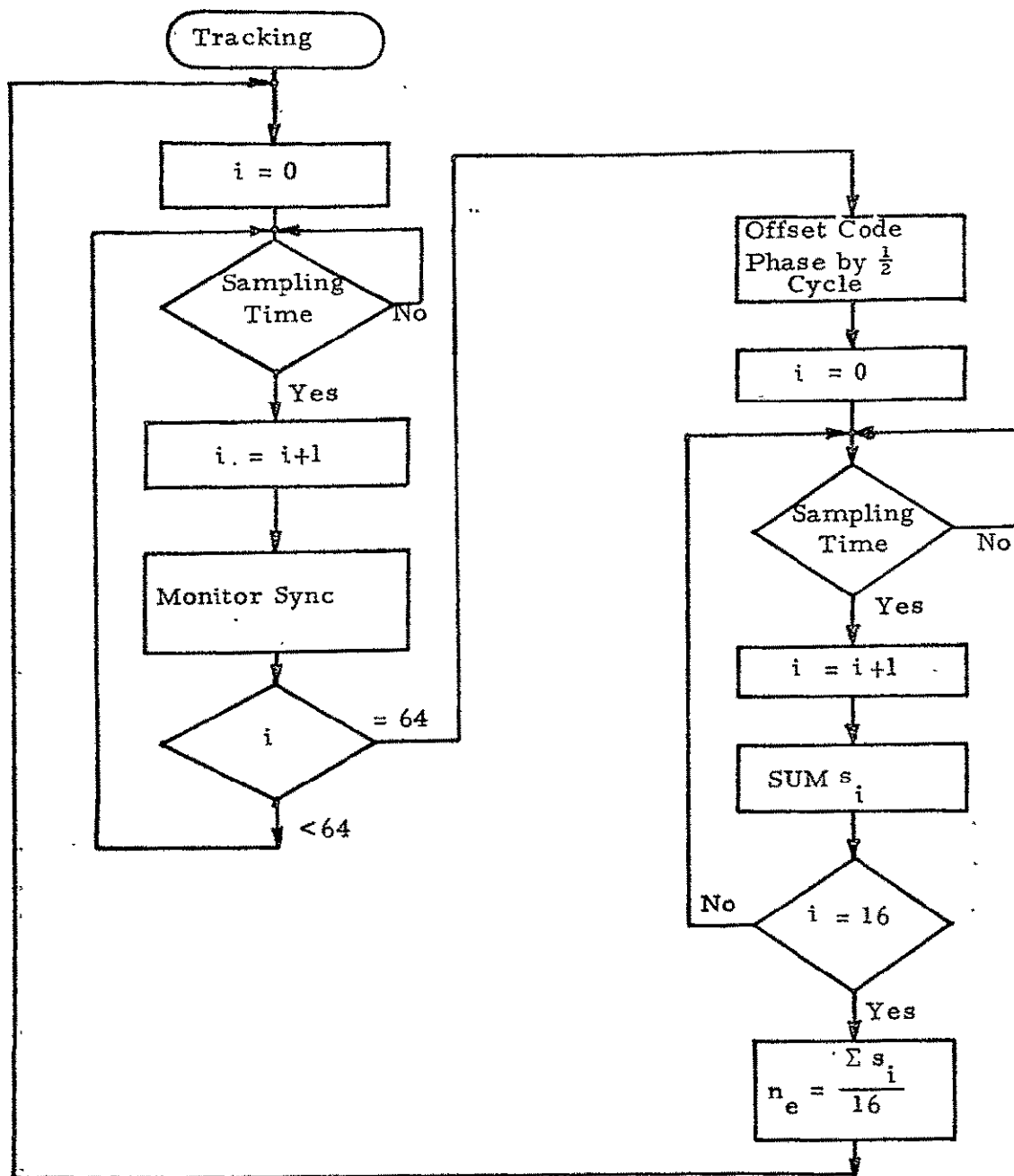
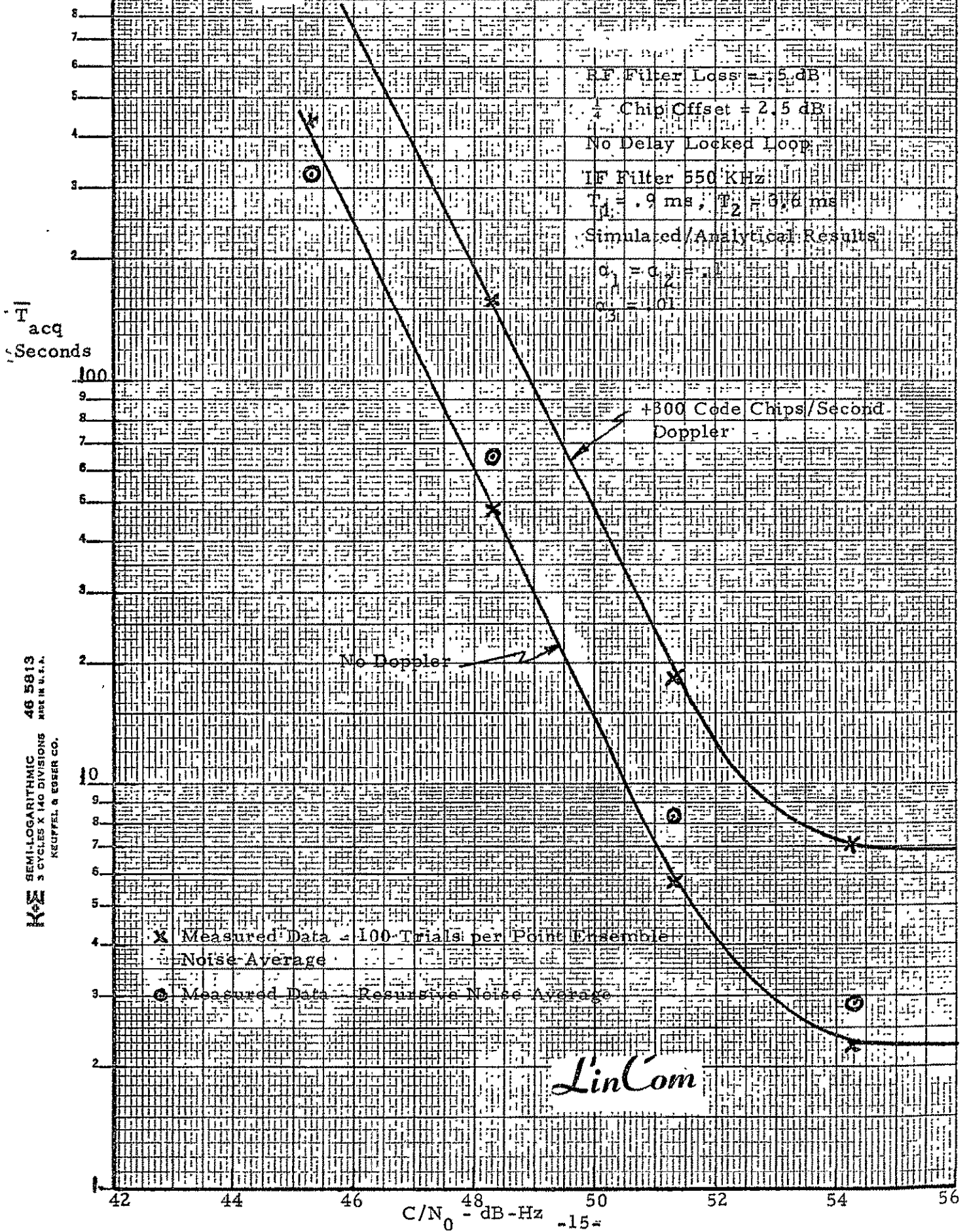


FIGURE 2.6. TRACKING NOISE ESTIMATE CONCEPT.

K&W
 SEMI-LOGARITHMIC
 46 5813
 3 CYCLES X 140 DIVISIONS
 MADE IN U.S.A.
 KEUFFEL & ESSER CO.

FIGURE 2.7. ACQUISITION TIME VS C/N



where Δf is the doppler offset and T_D is the equivalent dwell time at each code position and which for the search algorithm selected is given by

$$T_D = T_1 + \alpha_1 T_1 + \alpha_1^2 T_2 + \alpha_1^2 \alpha_2^2 24.5 T_2.$$

In the above equation α_1 is the probability of falsely indicating the presence of sync with signal absent on either hit in acquisition α_2 is the probability of falsely detecting sync on the single hit in verify and $24.5 T_2$ represents the average time it takes to discover that the system has falsely entered the tracking mode. The parameters T_1 and T_2 are as defined in section 2.1.1. The probability p is given by

$$p = (1 - \beta_1)^2 (1 - \beta_2)$$

where β_1 is the probability of missing sync with the signal present in acquisition and β_2 is the probability of not verifying the occurrence of sync. For the theoretical curves of Figure 2.7, $\alpha_1 = .1$, $\alpha_2 = .01$, $T_1 = .9$ ms and $T_2 = 3.6$ ms. The corresponding threshold settings were TH 1 = 1.056, TH 2 = 1.028 and TH 3 = 1.051. In generating these curves a .5 dB loss was allocated to the RF filter front end and a single hit per code cycle was assumed; the hit occurring when the relative code phases were offset a $\frac{1}{4}$ chip (2.5 dB fixed loss). The effect of the recursive filter on the noise estimate and the true delay lock loop behavior are not reflected in these curves.

2.2 Delay Lock Loop Hardware

A τ -dither delay lock loop (shown in Figure 2.8) has been selected for simulation. This implementation has poorer theoretical performance than the more conventional types of delay lock loops, but is less complex from a hardware point of view and is also less sensitive to equipment degradations due to the commonality

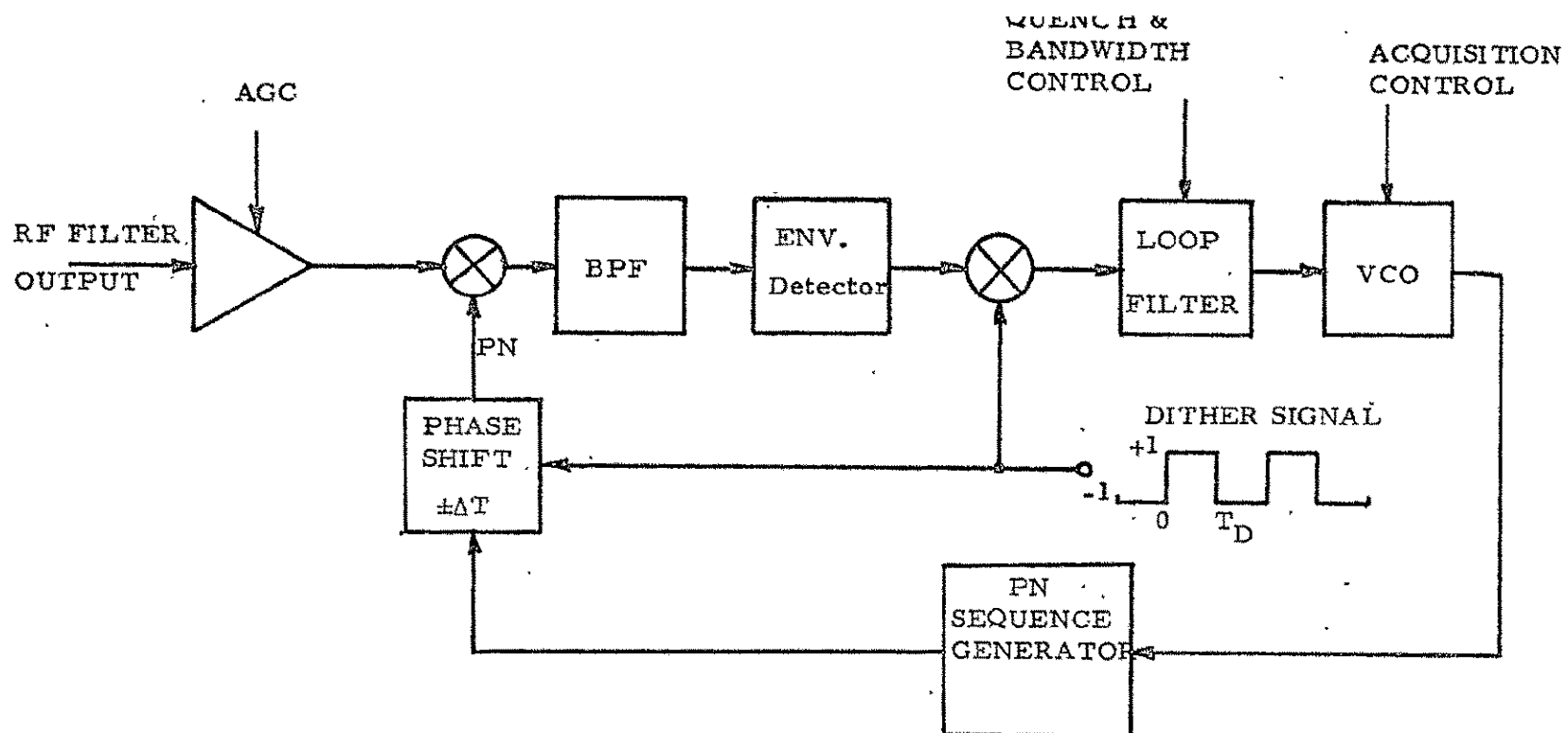


FIGURE 2.8. τ -DITHER DELAY LOCKED LOOP.

LinCon

of circuits. A dither signal, shown in the figure, shifts the PN generator $\pm\Delta T$ from its nominal code phase, which in turn provides early and late estimates of the correlation voltage. The polarity of the correlation voltage entering the loop filter is also switched at the dithering rate. The net effect is to produce a low frequency signal that drives the VCO until the early and late correlation voltages are of equal magnitudes. In the absence of RF front end and transmission degradations this condition maximizes the peak correlation voltage, hence the signal levels, seen by the PN acquisition, carrier recovery, and bit synchronization hardware.

A noncoherent detection of the correlated signals, consisting of a bandpass filter and envelope detector demodulate the early/late correlation voltages. A 550 KHz, 2 pole low pass equivalent, Butterworth bandpass filter is used. The static offset resulting from code doppler is reduced to zero by using a loop filter with transfer function $(\tau_2 s + 1)/\tau_1 s$. In acquisition the τ -dither loop noise bandwidth is 200 Hz with a .707 damping factor for an input C/N_0 of 48.3 dB. The charge on the loop filter capacitor is quenched by the microprocessor each time a move to a new code cell position commences. The code phase is dithered a half-chip in either direction from nominal at a rate equal to a quarter of the IF filter bandwidth ($BT_D = 4$).

2.2.1 Loop Behavior

Loop jitter versus C/N_0 is plotted in Figure 2.9. Note the large amounts of jitter at the C/N_0 values of interest. The transient response of the loop to a $\frac{1}{4}$ chip step change in phase plus a 300 code chip doppler offset at an input C/N_0 value of 48.3 dB-Hz is plotted in Figure 2.10A. Even though the loop can respond and track doppler the randomness

LinCom

$B_{IF} = 550 \text{ KHz}$

RF Filter Loss = 1.5dB

$B_{IF} = 200 \text{ Hz}$

$\xi = .707$

$B_{IF/D} = 4$

σ_{ϕ} CHIPS

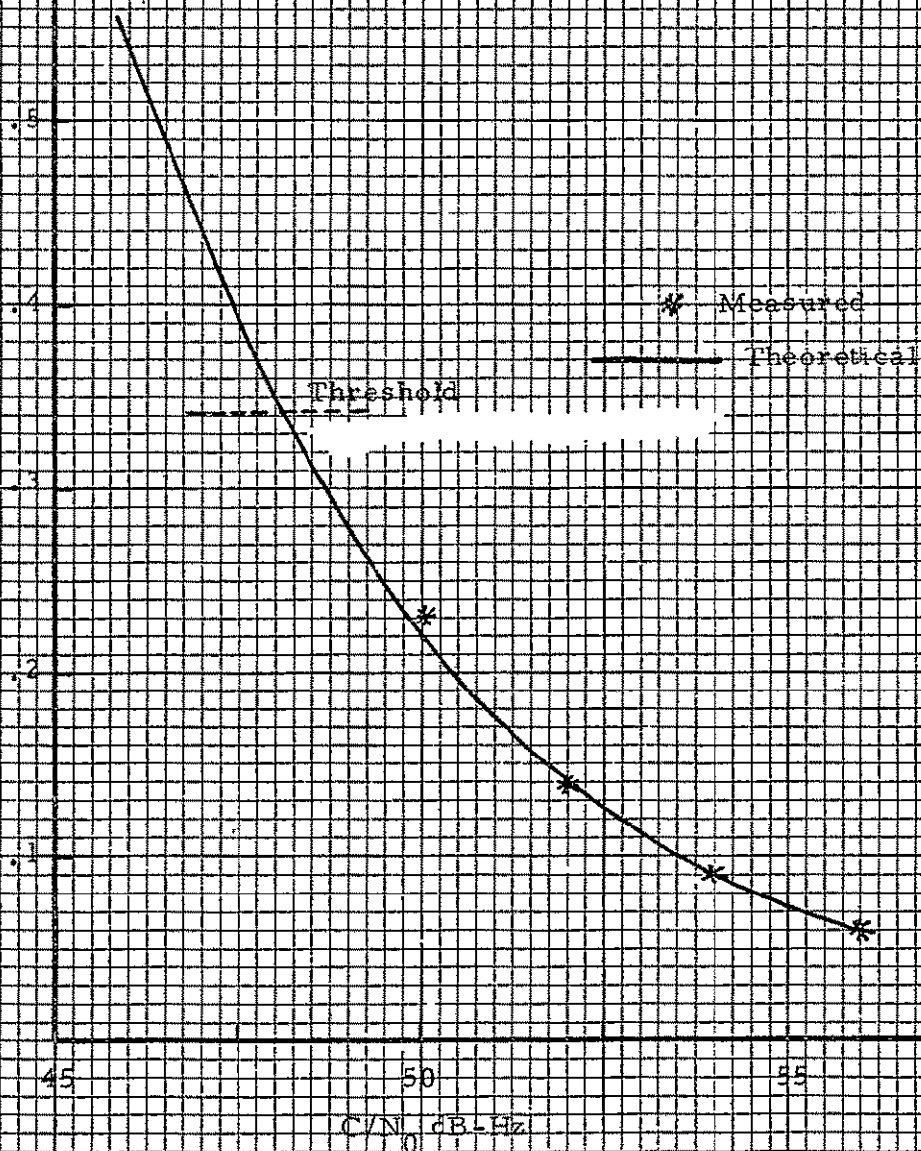


FIGURE 2.9. RMS PHASE ERROR VERSUS C/N_0

FIGURE 2.10. CODE ACQUISITION TRANSIENT DURING LOOP BANDWIDTH

REDUCTION.

Ideal RF Filter

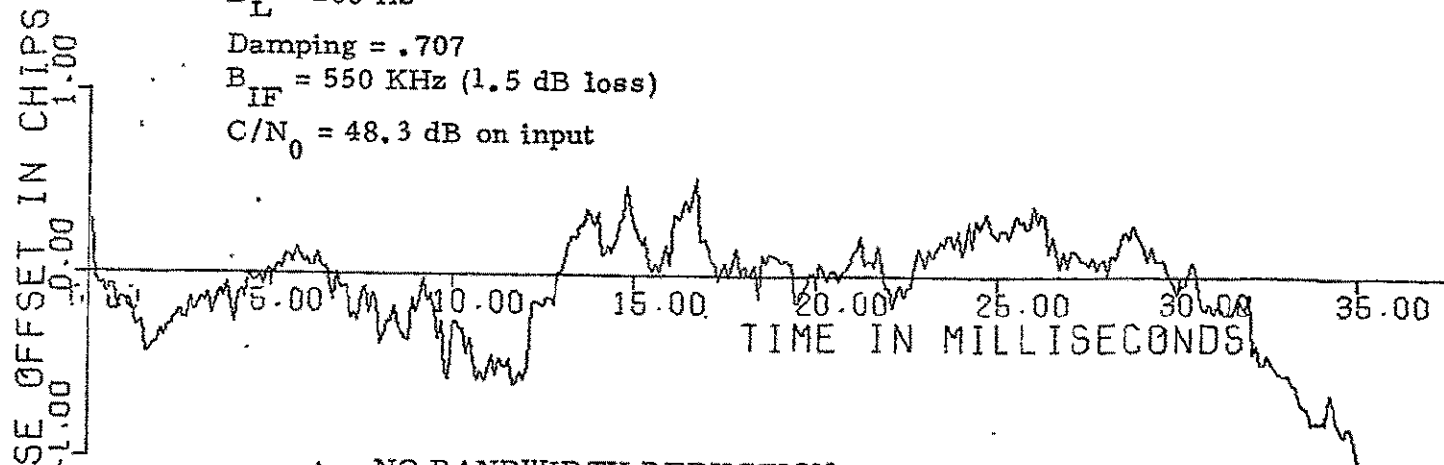
Code Doppler = -300 chips/sec

$B_L = 200$ Hz

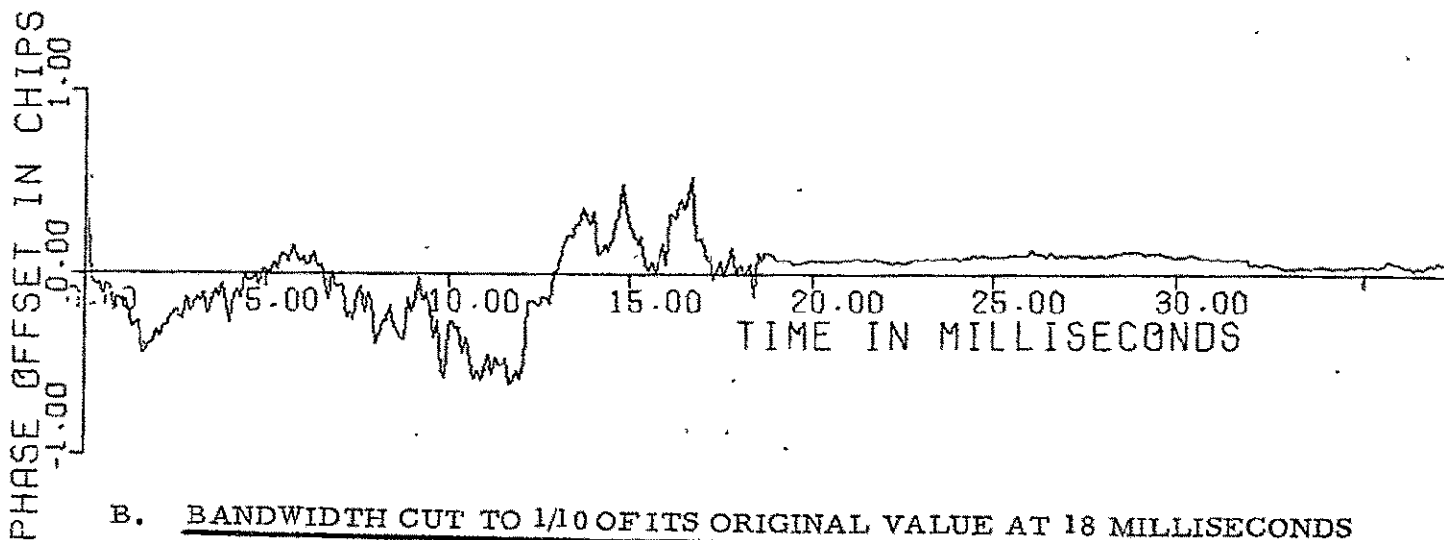
Damping = .707

$B_{IF} = 550$ KHz (1.5 dB loss)

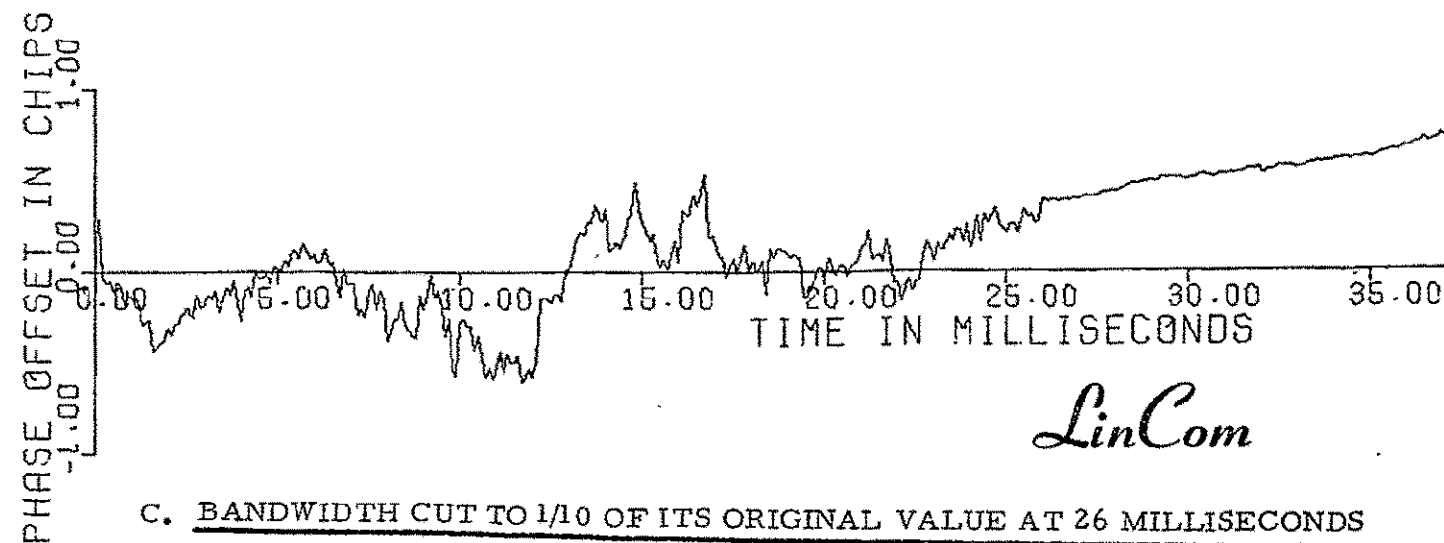
$C/N_0 = 48.3$ dB on input



A. NO BANDWIDTH REDUCTION



B. BANDWIDTH CUT TO 1/10 OF ITS ORIGINAL VALUE AT 18 MILLISECONDS



C. BANDWIDTH CUT TO 1/10 OF ITS ORIGINAL VALUE AT 26 MILLISECONDS

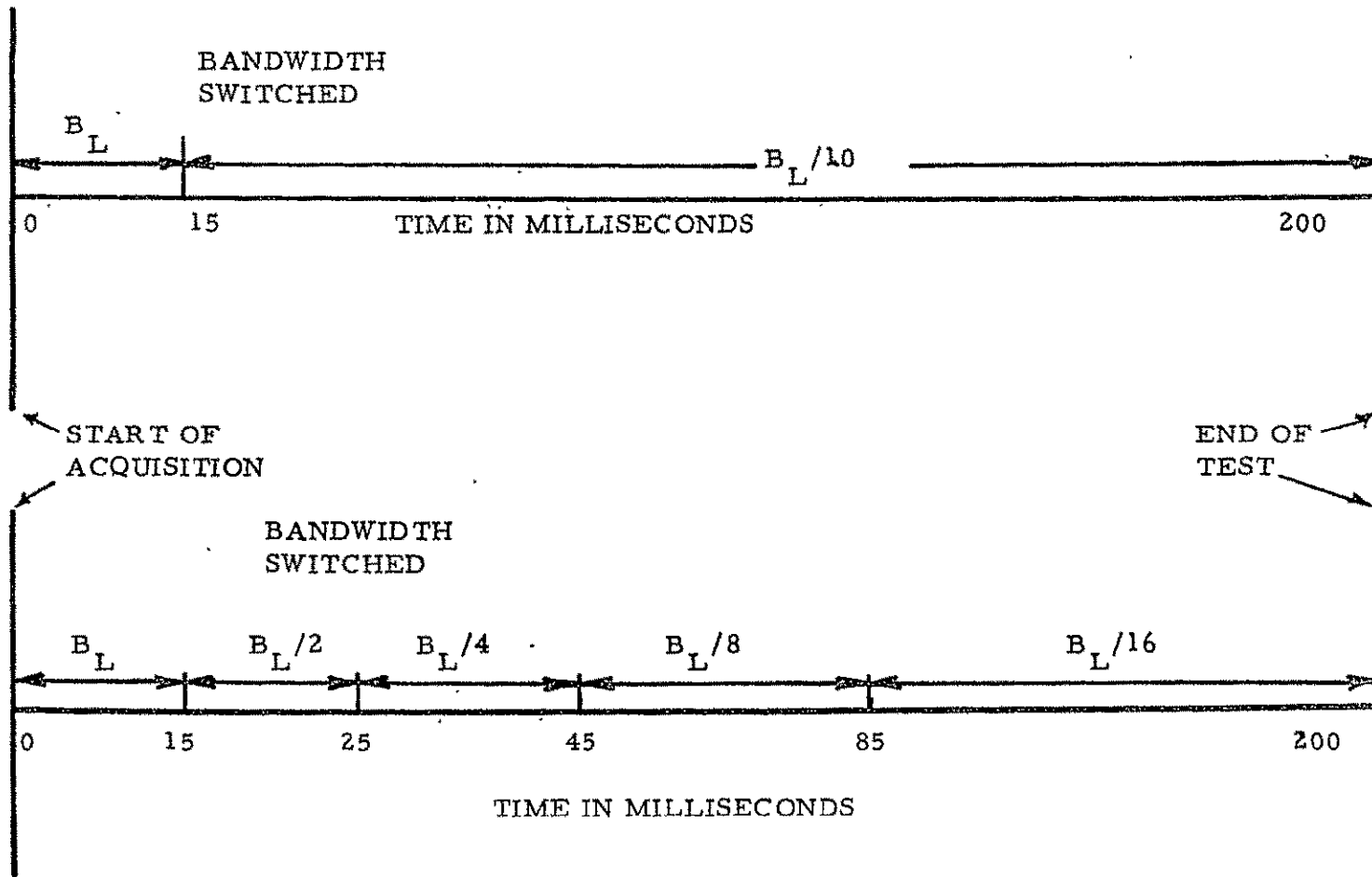
LinCom

of the noise eventually walks the system out of lock. For the case shown this occurs at about 35 milliseconds and highlights the need for bandwidth reduction, once PN sync has been found. Figure 2.10B illustrates the behavior of the loop with the bandwidth cut suddenly by a factor of 10. This occurs after approximately 18 milliseconds. Note how the loop quiets down and slowly drifts to a very low phase error. However, if one switched bandwidths at 26 milliseconds instead of 18, as shown in Figure 2.10C, note what happens. The loop goes out of lock with a response time set by the lowered bandwidth, i.e., very slow but steady. The magnitude of the transient response due to switching bandwidths, depends upon the signal to noise ratio, the magnitude of the bandwidth change, and most importantly the charge on the loop filter at the switching instant. In Figure 2.10B the loop was switched at the instant when the slowly varying filter capacitor voltage drifted through its mean value. In Figure 2.10C the loop was intentionally switched when the capacitor voltage, tracking the in-band noise, drifted .2 volts higher than the mean. Figure 2.10A demonstrates the requirement for bandwidth switching in a system such as ours which operates in deep noise. Figures 2.10B and 2.10C serve as best and worst case examples of the effects of such a switching on loop behavior.

2.2.2 Bandwidth Reduction

An experiment was performed to evaluate two bandwidth reduction techniques. The experiment consisted of repeatedly activating the τ -dither loop with an initial .25 chip phase offset and 300 code chips of doppler and evaluating its behavior as a function of bandwidth reduction algorithm and signal to noise. The two algorithms used are shown in Figure 2.11. Algorithm #1 abruptly reduced the

ALGORITHM #1 - ABRUPTLY REDUCING BANDWIDTH BY A FACTOR OF TEN



ALGORITHM #2 - STEPPING BANDWIDTH DOWN 4 TIMES

FIGURE 2.11. BANDWIDTH REDUCTION ALGORITHMS.

LinCom

bandwidth to a tenth its initial value 15 milliseconds after the start of acquisition. Algorithm #2 halves the bandwidth 15 milliseconds after the start of acquisition, waits 10 milliseconds more and again reduces the bandwidth by a factor of 2, waits 20 more milliseconds and again halves the bandwidth, waits another 40 milliseconds and cuts the bandwidth to a final value equal to $1/16$ of its initial value. The initial bandwidth was 200 Hz at a C/N_0 of 48.3 dB-Hz with a .707 damping factor. This corresponds to a 333 Hz bandwidth and a damping factor of 1 at the system design point C/N_0 of 51.3 dB-Hz. During the stepping procedure the damping factor remained unchanged from its initial value.

Twelve trials were made at each of two C/N_0 values, for each of the two algorithms. Identical noise samples were used during each trial for testing all 4 combinations of C/N_0 and algorithm. The noise from trial to trial differed. This allows one to make a relative evaluation of performance without excessive amounts of testing. The results of the experiment are tabulated in Table 2.1. The superiority of stepping the bandwidth 4 times as opposed to abruptly can be seen from the data. In either case the bandwidth reduction does not completely remove the possibility of walking out of sync. It just reduces the probability of it happening. The superior algorithm of stepping the bandwidth 4 times instead of abruptly was selected for implementation in the system simulation. In acquisition the loop noise bandwidth is 200 Hz with a .707 damping factor and is reduced in four steps to 12.5 Hz in tracking.

2.3 Loops Impact on Acquisition Time

The theoretical curve of T_{acq} vs C/N_0 of Section 2.1.3 is redrawn as curve #1 in Figures 2.12, 2.13 and 2.14 for the conditions of 0, +300 and -300 code chips/sec of doppler. As previously stated these were generated at the subsystem level with a flat 2.5 dB loss allocated.

C/N ₀ dB-Hz	Number of Trials Loop Walked Out of Sync Prior to First 15 ms	Number of Trials Loop Walked Out of Sync Prior to 200 ms Test Duration	Number of Trials Sync Would be Eventually Lost (Predicted)	Condition
48.3	3	8	12	No Bandwidth Reduction
51.3	1	0	12	Trials = 12
48.3	3	9	9	Bandwidth Cut by Ten
51.3	1	2	2	Trials = 12
48.3	3	3	3	Bandwidth Stepped 4 Times
51.3	1	2	2	Trials = 12

TABLE 2.1. BANDWIDTH REDUCTION EXPERIMENT.

LinCom

1000

FIGURE 2.12 ACQUISITION TIME VS C/N₀T_{acq}
Seconds

100

9

8

7

6

5

4

3

2

10

9

8

7

6

5

4

3

2

1

42

44

46

48

50

52

54

56

C/N₀ dB-Hz

-25-

No Doppler

Chebyshev RF Filter (4-Pole)

IF Filter 550-KHz

T₁ = 0.9 ms, T₂ = 3.6 msα₁ = α₂ = 1, β₁ = β₂ = 0.1B₁ = 200 Hz @ 48.13 dB C/N₀

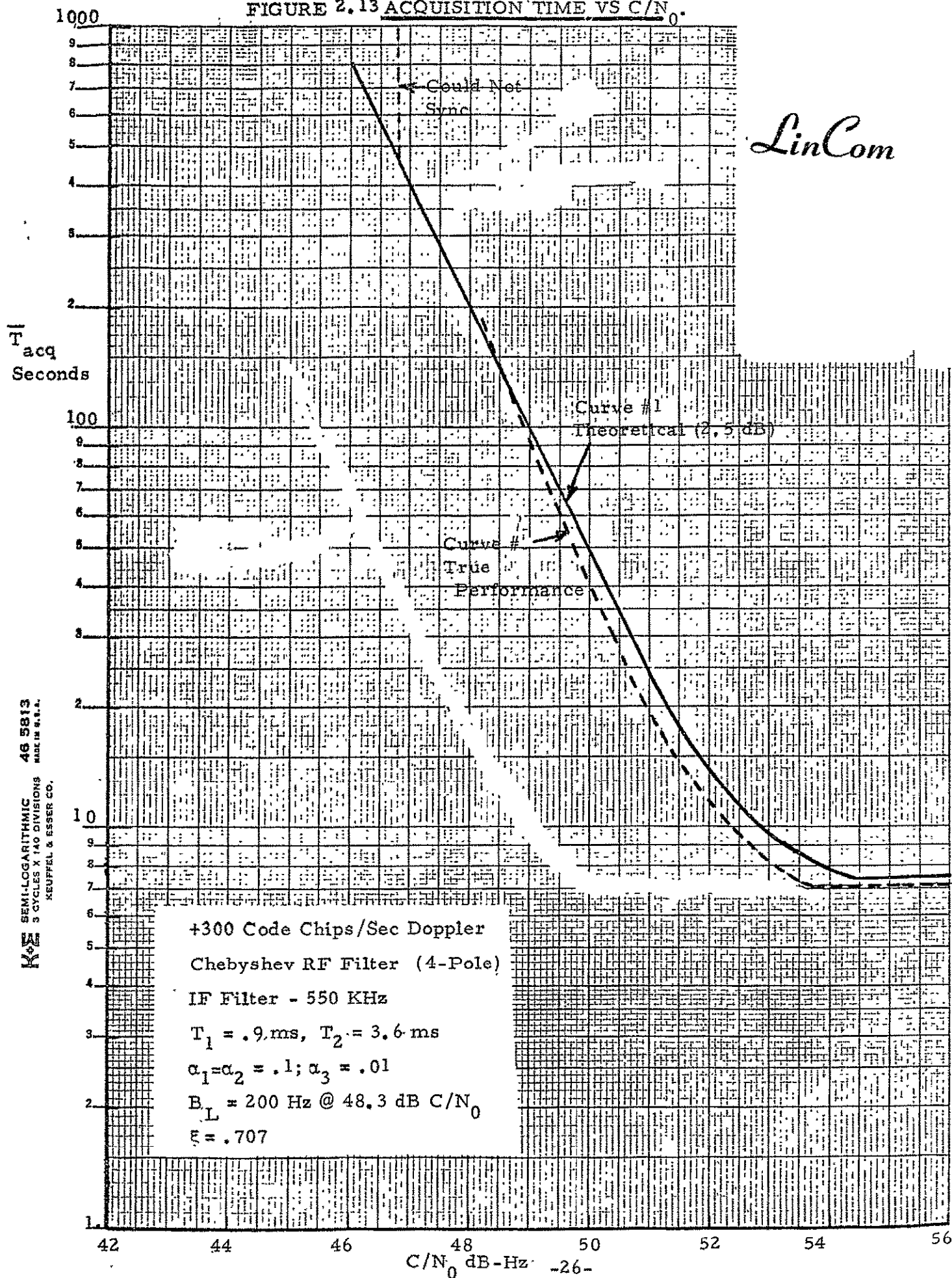
Loss = 70%

Curve #1
Theoretical
(2.5 dB Loss)Curve #2
True Performance

LinCom

46 5813
SEMI-LOGARITHMIC
3 CYCLES X 140 DIVISIONS
MADE IN U.S.A.
KEUFFEL & ESSER CO.

FIGURE 2.13 ACQUISITION TIME VS C/N_0



1000

FIGURE 2.14. ACQUISITION TIME VS. C/N

-300 Code Chip/Sec Doppler

Chebyshev RF Filter (4-Pole)

IF Filter - 550 KHz

 $T_1 = .9 \text{ ms}, T_2 = 3.6 \text{ ms}$ $\alpha_1 = \alpha_2 = .1; \alpha_3 = .01$ $B_L = 200 \text{ Hz @ } 48.3 \text{ dB } C/N_0$ $\xi = .707$ T_{acq}
Seconds

100

Curve #1
Theoretical (2.5dB)

LinCom

Curve #2
True
Performance

10

9

8

7

6

5

4

3

2

1

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

to the degradation associated with τ -dither loop hardware. The true overall acquisition performance differs from these curves. The effects of the loop jitter and transient characteristic during the locking interval will degrade performance as a function of C/N_0 and code doppler. However this performance loss will be offset by the fact that more than one chance to acquire sync exist during a single sweep and that the initial phase is random in nature and not the fixed .25 chip worst case offset (2.5 dB loss) assumed in generating curve #1. The actual impact that the loop has on performance is plotted as curve #2 in the three figures. At all but low C/N_0 values actual performance is better than that predicted by curve #1. At C/N_0 values below about 48 dB behavior becomes worse than that predicted by curve #1.

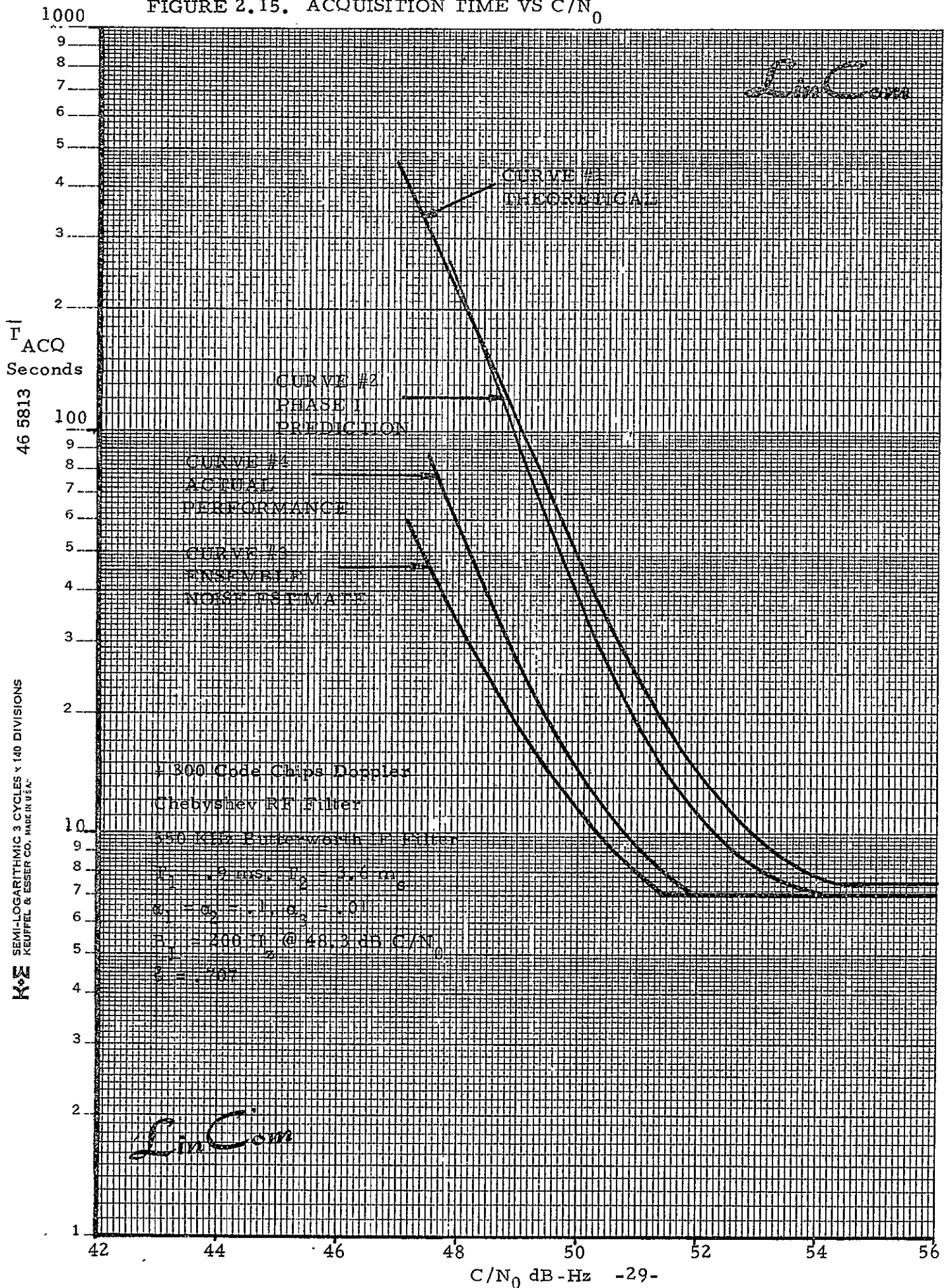
As stated in the Phase I report (Section 3.6.1) curve #2 of Figures 2.12 and 2.14 come close to representing the true overall performance of the PN Subsystem. These curves do not however reflect the increased mean acquisition time associated with the loop walking out of sync prior to completion of the bandwidth reduction process. Curve #2 of Figure 2.13 does not represent true system behavior in the presence of positive doppler.

2.3.1 T_{acq} Significantly Reduced in Positive Doppler

The curves of Figure 2.13 are reproduced in Figure 2.15. Curve #1 represents the theoretical prediction of acquisition times and curve #2 the simulated results including τ -dither loop behavior. Curve #2 data was measured by counting the number of chances to sync, N_c , and the number of time sync occurred, N_s . For zero or negative doppler this technique is quite valid and T_{ac} is given by

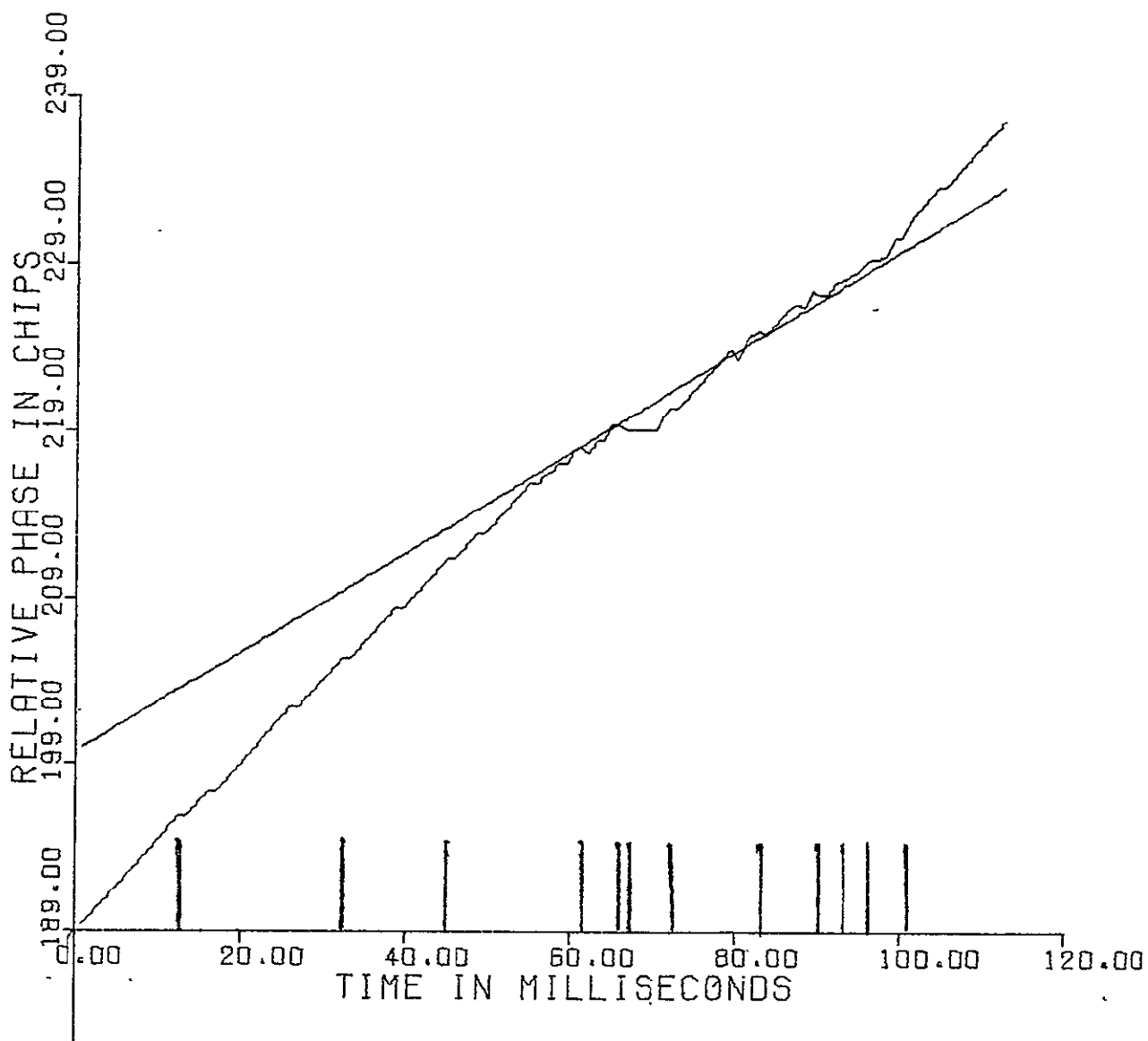
$$T_{acq} = T_M \left(\frac{2N_s}{N_c} - 1 \right),$$

FIGURE 2.15. ACQUISITION TIME VS C/N_0



where T_M is the mean acquisition time in high signal to noise. To account for positive doppler N_c and N_s must be replaced by N'_c , the number of code cycles measured and N'_s , the number of times sync occurred within a code cycle. With the simulation modified to measure N'_c and N'_s predicted performance is shown by curve #3. The significant performance improvement will be explained with the aid of Figure 2.16. This figure displays curves of code phase (relative to a nominal PN code generator with its clock set at exactly the chip rate versus time. The smooth curve represents the incoming PN pattern offset by +300 Hz of code doppler with time zero adjusted to produce an initial phase position of ~199 chips measured from the stationary reference. The other curve represents relative phase of the locally generated replica which is being stepped by microprocessor command and jittered via delay lock loop action. The spike marks along the bottom represent microprocessor sync indications. Prior to ~50 milliseconds these are indications of false sync as the code phases are not aligned. At about 60 milliseconds the local code phase has been stepped into the region where sync is possible. Now notice what happens. A number of first hit indications occur. At each of these stepping ceases, with the microprocessor searching for additional hits before true sync can be verified. At a C/N_0 of 48.3 dB-Hz going all the way through verification has a low probability of occurrence and thus when sync is not declared resetting of the local code phase occurs and the search continues. Meanwhile the transmitted code phase has been advancing due to doppler (minimum increase slightly over .5 chips). The net effect is to produce a region lasting over many chip intervals where the local phase is trying to catch up to the transmitted code phase. In the figure a total of 7 chances to sync occurred before the local code advanced past the transmitted one and sync was no longer possible on this particular PN cycle.

FIGURE 2.16. CODE PHASE VERSUS TIME.



Experiments conducted in generating data for Figure 2.16 and curve #3 of Figure 2.15 used an ensemble estimate of noise power.

2.3.2 Recursive Filter Noise Estimate Effect on T_{acq}

The noise estimate algorithm used in our simulation was described in Section 2.1.2. Curve #4 of Figure 2.15 includes its effect on acquisition performance for positive doppler. Degradation attributed to this algorithm can be measured by comparing curve #3 and curve #4. An explanation for this reduction in performance will now be presented with the aid of an experiment designed to observe interactions between the loop, sync detector voltage, and noise estimate. Figure 2.17 displays integrate and dump voltage used for sync detection and threshold times noise estimate as a function of time. Pn code search is assumed to commence at time zero. The figure starts at 21.89 seconds with the microprocessor having stepped our local code phase to near PN alignment. Up to about 21.93 seconds the sync voltage represents noise which from a close study of Figure 2.17 periodically exceeds threshold. These false sync indications are ignored as sync fails to be verified. Note the correlation between signal and noise estimate during this time interval. Large variations in output voltage visibly changing our noise estimate. At about 21.93 seconds code alignments occurs and sync is detected and verified. Note the drop in threshold times noise estimate which occurs during verify and is caused by a lowered threshold value (TH 2). Immediately after sync is verified the loop bandwidth starts to narrow and a higher threshold TH 3 is selected. TH 3 causes the noise estimate waveform to jump back up where it remains unchanged until about 22 seconds. Unfortunately during bandwidth switching sync was lost as can be observed by the drop in output voltage below the noise estimate. It takes until 22 seconds for loss of sync to be detected, at which

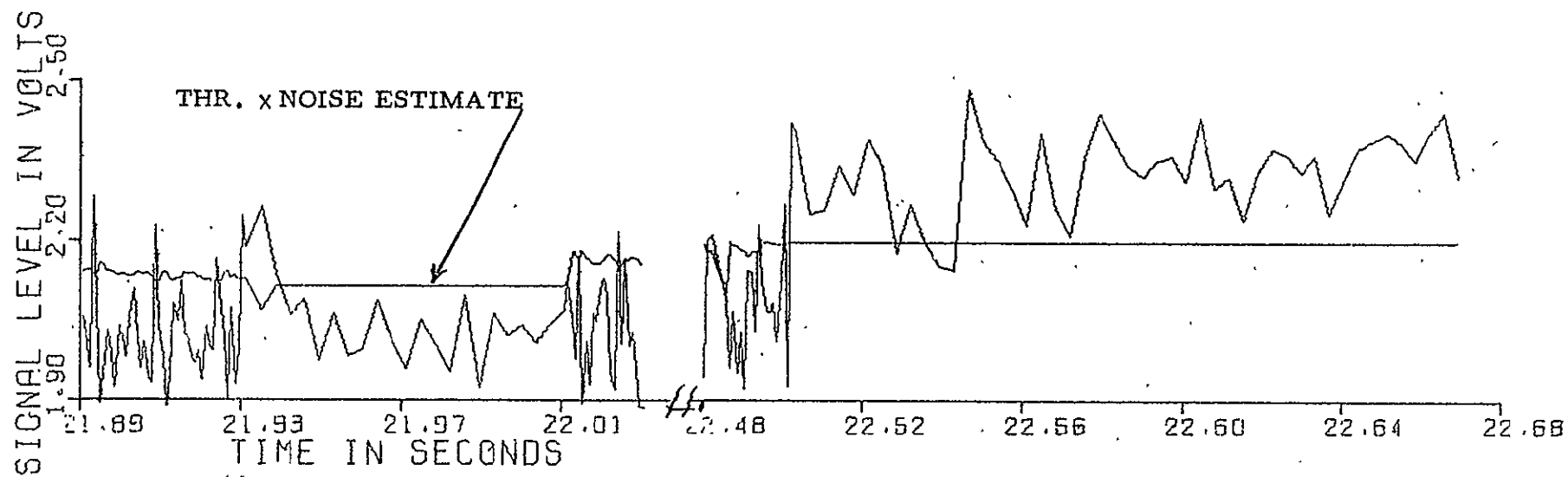


FIGURE 2.17. INTEGRATE AND DUMP OUTPUT VERSUS TIME, $C/N_0 = 48.3$, 300 Chips/sec.

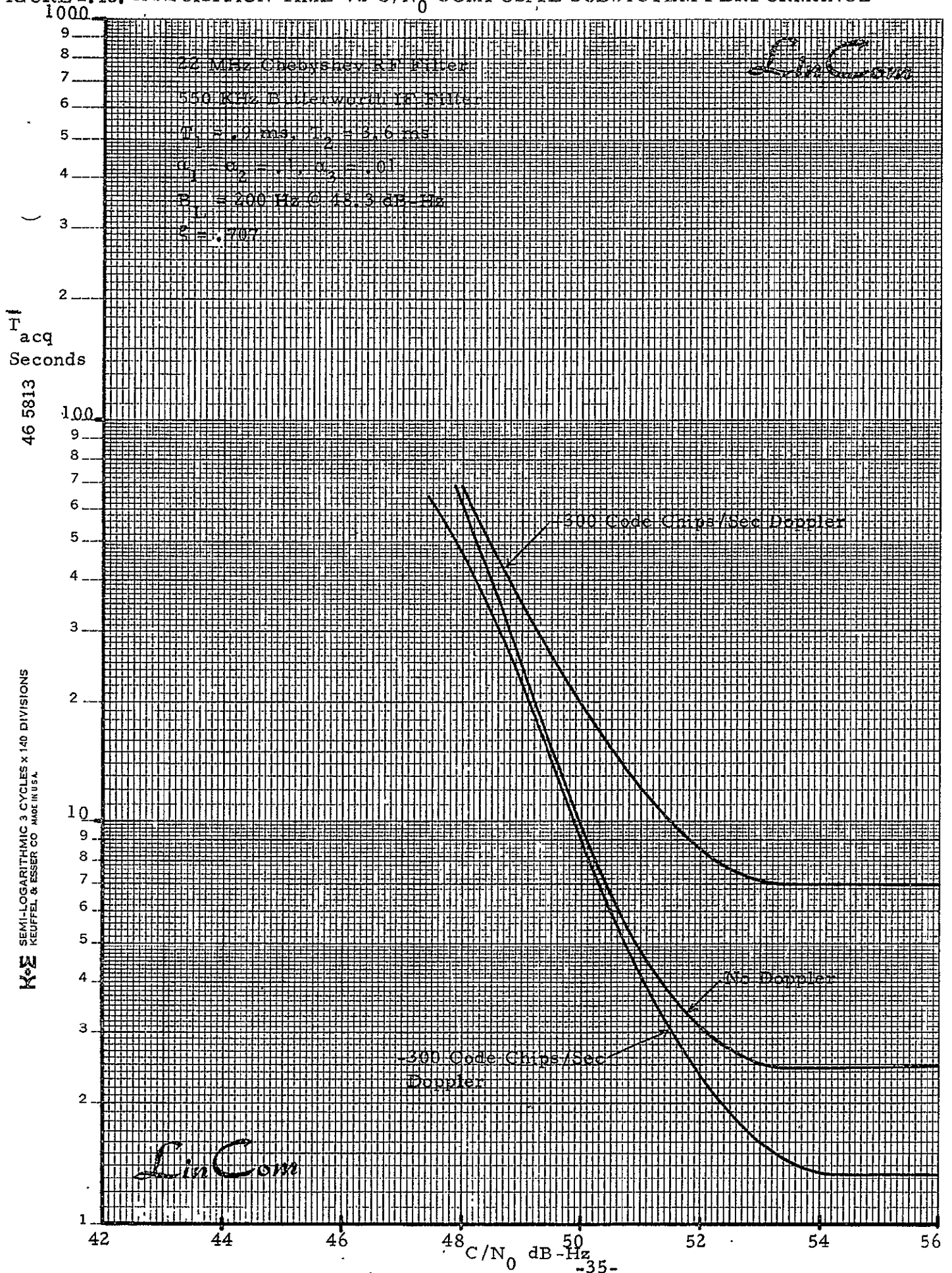
point our noise estimate is updated and the search continues. Note an increased threshold times noise estimate after update, the new estimate being contaminated by the correlated signal.

Between 22 and 22.48 seconds the code phases stay nearly aligned by the action described in section 2.3.1. Each time a hit occurs but is not verified the updated noise estimate is biased upward due to signal correlation, making it increasingly difficult for sync to be verified. This lifting of our threshold within the sync region causes the degradation observed in curve #4 of Figure 2.16. In Figure 2.17 sync is finally achieved after 22.48 seconds even with the relatively large noise estimate shown.

2.4 PN Subsystem Integrated Performance

Curve #2 of Figures 2.12 and 2.14 and curve #4 of Figure 2.16 come close to representing true overall acquisition performance at the subsystem level. They include effects of loop behavior, noise estimate and sync algorithms, RF and IF filter degradations and doppler. One additional factor, loop bandwidth reduction, is still to be considered. Data listed in Table 2.1 indicates a 25% and 16% probability at respectively 48.3 and 51.3 dB-Hz, that sync will be lost during our 4 step bandwidth switching process. True overall predicted acquisition performance which includes bandwidth reduction behavior is shown in Figure 2.18.

FIGURE 2.18. ACQUISITION TIME VS C/N_0 COMPOSITE SUBSYSTEM PERFORMANCE



SECTION 3

PN SUBSYSTEM - NASA DESIGN

3.0 Introduction

The PN Subsystem discussed in Section 2 was based on a LinCom design developed during our Phase I simulation effort. Concurrent with this effort, Harris, Inc. designed and constructed an actual hardware despreaders which is identical to that simulated except for a series of sync algorithms. This section describes a stand-alone PN despreaders simulation software package, incorporating the algorithms employed by Harris. It is intended for use by NASA in supporting evaluation and testing of the actual hardware.

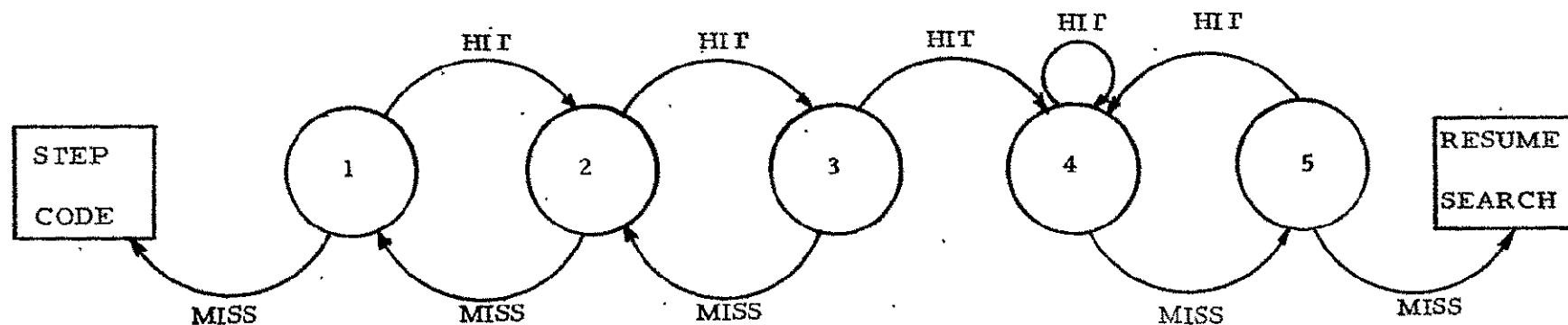
3.1 Sync Algorithms

The algorithms shown in Figure 3.1 have been programmed into the simulation. These algorithms were obtained from an internal NASA paper outlining the proposed testing of the ESTL spread spectrum modem.

3.2 Software

Two independent software programs are required for simulating the actual hardware. The first of these generates and stores on disk a large number of samples of the PN acquisition hardware output (see Figure 2.2). This program is run only when the IF filter bandwidth or the basic search state integration time (K in Figure 3.1) is varied. The second software program, employing Monte Carlo type techniques, use these stored samples to predict ESTL modem behavior. A basic software restriction, one also placed on the hardware, is that the check and lock mode integration times be multiples of that of the search mode.

LinCom



SEARCH

ONE STAGE

THRESHOLD: D_S

TIME: $K/32000$

CHECK

N_C STAGES

THRESHOLD: D_C

TIME: $KS_C/32000$

LOCK

N_L STAGES

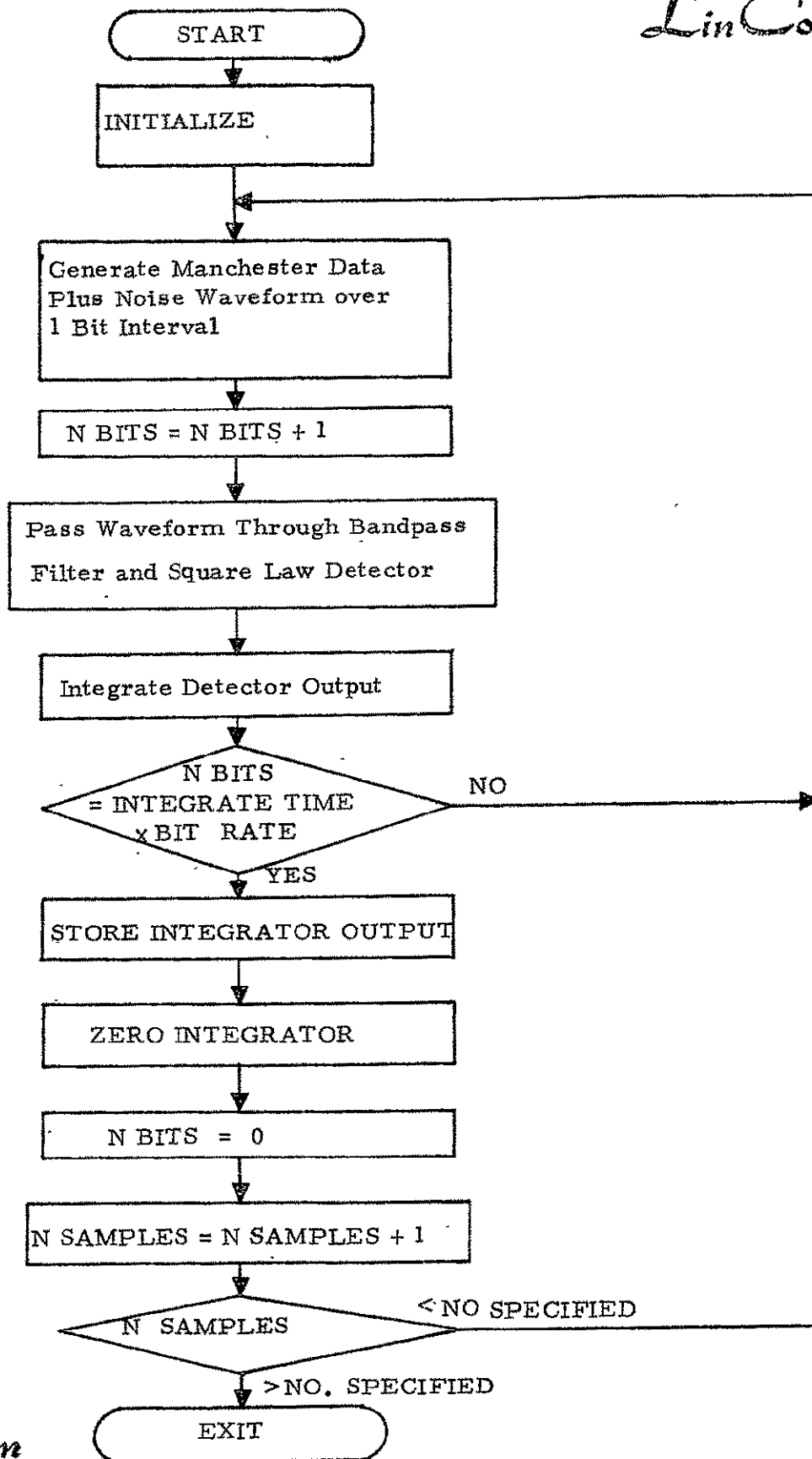
THRESHOLD: D_L

TIME: $KS_L/32000$

FIGURE 3.1... SIMPLIFIED UP-DOWN COUNTER LOGIC USED IN HARRIS DESPREADER AND IMPLEMENTED IN THE SIMULATOR.

LinCom

FIGURE 3.2. PN ACQUISITION OUTPUT SAMPLE GENERATION.



3.2.1 Program Generating PN Acquisition Output Samples

A high level flow chart describing this program's operation is shown in Figure 3.2. A more detailed flow chart and program listings are to be found in Appendix A.

3.2.1.1 Input Parameters

A single data card is used to input all parameters via the following call:

```
READ 11, NTIME, NXX, NYY, BITRAT, BIF2
```

In the above NTIME is the number of samples per bit, NXX is the number of bits integrated over (equal to the integration time multiplied by the symbol rate), NYY is the number of integrate and dump outputs stored, BITRAT is the symbol rate, and BIF2 is the IF filter noise bandwidth in Hz. The following format must be used to input this data:

```
11 FORMAT(3I10, 2F10.2)
```

Please note that signal to noise ratio is not an input parameter. It is however accounted for within the program.

3.2.1.2 Output Data

The Integrate and dump output sample for 19 values of C/N_0 are written on file #3 with the following statement:

```
WRITE(3) Y .
```

In addition, all values of Y for all samples are printed out, along with the input data read.

3.2.2 PN Despreader Simulation Program

For programming purposes, the despreader hardware is partitioned into 4 major elements as follows:

```
Noncoherent Correlation/ - SUBROUTINE PNACQ  
Delay Lock Loop Interface  
τ-Dither Loop and RF Filter - SUBROUTINE LOOP  
I & D Output Generation - SUBROUTINE THRESH  
Microprocessor - SUBROUTINE PNMON
```

At each decision time subroutine PNACQ obtains a random sample representing an I & D output value modified by the loop action via calls to subroutines Loop and THRESH. It then passes this information to the microprocessor via a call to subroutine PNMON. Subroutine LOOP simulates the effects of τ -dither delay lock loop action on the correlation voltage. Information from LOOP is passed to subroutine THRESH which randomly selects an I & D output sample from a file that has been generated earlier during the execution of the program described in section 3.2.1. This sample is then modified by our desired signal to noise and the information generated in subroutine LOOP and passed back to subroutine PNACQ. The above mentioned interactive process is shown in flow chart form in Figure 3.3. A more detailed set of flow charts and program description can be found in Appendix A.

3.2.2.1 Input Parameters

Five groups of parameters are inputted via data cards. The first data block, consisting of the following three cards and associated parameters is called from the main program.

CARD 1: CODDOP, BITRAT, BWPNAC, RFBW

CARD 2: THP1, THP2, THP3

CARD 3: TS1, TS2, TS3

In the above CODDOP is the code doppler in chips per second, BITRAT is the symbol rate in symbols per second, BWPNAC is the Acquisition Hardware IF bandwidth in Hz, and RFBW is the RF bandwidth in Hz. THP1, THP2, and THP3 are the search, check, and lock threshold parameter and correspond directly with the hardware thumb wheel settings labeled D_S , D_C , and D_L (see Figure .1). TS1, TS2, and TS3 are the search, check, and lock integration times in seconds and correspond to $K/32000$, $KS_C/32000$, and $KS_L/32000$ where S_C and S_L .

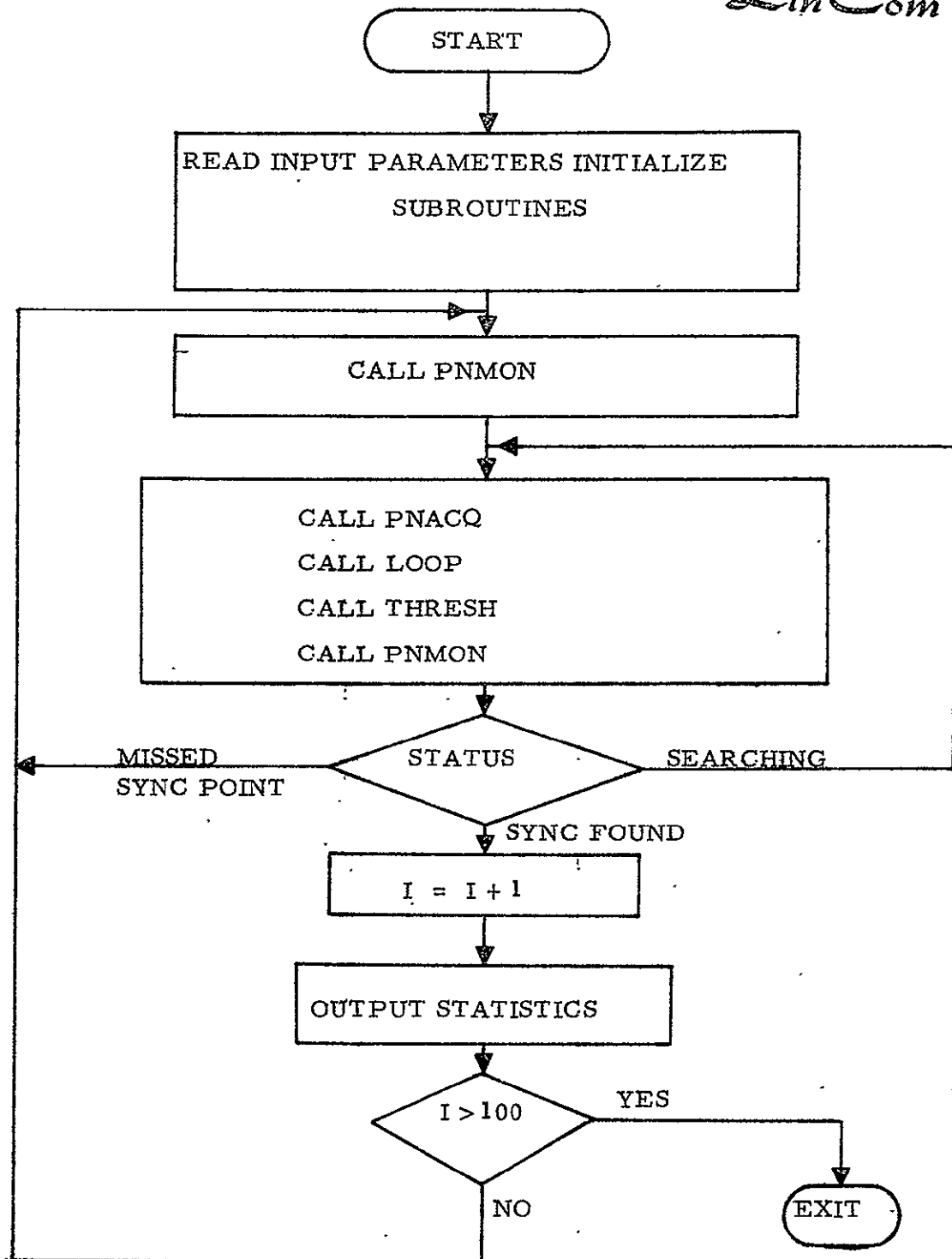


FIGURE 3.3 HIGH LEVEL FLOW CHART OF PN DESPREADER PROGRAM.

are integers greater than 0. Unlike the hardware, $K/32000 = TS1$ should be an integer multiple of BITRAT if exactness is desired.

The second data block consisting of the following single card and parameters is called by subroutine PNMON.

CARD 4: NOEST, NJPCYL, IOPT9, TINC9, NCHCL, NLKCL

In the above NOEST is the noise estimate option (=0 for recursive noise estimate algorithm, =1 for ensemble noise estimate), NJPCYL is the sweep/step option. If NJPCYL = 0 the program sweeps only through a narrow window centered about the true sync position and thus measures the probability of synchronizing on a single sweep, p . If NJPCYL = 1 the program continuously sweeps through all code cells and thus measures T_{acq} directly. To reduce computer execution time one would normally run the program at a high value of C/N_0 with NJPCYL = 1, thus evaluating T_{acq}^H . For low values of C/N_0 NJPCYL is set to zero and the acquisition time is evaluated using

$$T_{acq} = \left(\frac{2-p}{p} \right) T_{acq}^H .$$

IOPT9 is a loop option, with IOPT9 = 0 the loop is activated, if IOPT9 = 1 the loop is bypassed. TINC9 is the search window size, with TINC9 = 5 the window is narrowed to a $\pm .25$ chip interval allowing only a single chance to sync each sweep or step trial. Normally TINC9 is set equal to 10 corresponding to a $\pm .5$ chip window and thus allows for a multiple number of hits per trial. NCHCL is the number of stages in the check mode (corresponds to N_C of Figure 3.1). If NCHCL = 0 the check mode is bypassed. NLKCL is the number of stages in the lock mode (corresponds to N_L of Figure 3.1).

The third data block consisting of the following single data card and associated parameters is called by subroutine LOOP.

CARD 5: DLLZET, DLLBLL, CODDOP, CNO, DLLBIF,
ICOMM

DLLZET is the loop damping factor. DLLBLL is the loop noise bandwidth in Hz. The LOOP program automatically adjusts to give a DLLBLL Hz bandwidth at a 48.3 dB-Hz C/N_0 value. CODDOP is as previously defined. CNO is the input C/N_0 value. This value is degraded, internal to the program, by .5 dB to account for RF filter loss. DLLBIF is the τ -dither loop IF bandwidth in Hz. ICOMM is a noise option parameter. With ICOMM = 0 noise is added to the signal, with ICOMM = 1 loop operation is in the absence of noise.

The fourth data block consisting of the following single data card and associated parameters is called by subroutine THRESH.

CARD 6: ATTN, TS1, TS2, TS3, BWPNAC, IOPT

ATTN is the input C/N_0 value equal to CNO (see card 5). TS1, TS2, TS3, BWPNAC have been previously defined. IOPT is a loop option and must have the same value as IOPT9 which was previously described.

The last data block contains a single parameter on a single data card, IOPT, whose value is as defined above. The reader will note that a number of parameters have been duplicated on one or more data cards. This action has been taken for programming convenience and ease in disassembling these subroutines for other study efforts.

As a summary, exercising the program requires 7 data cards. Their order and formats are:

```
#1 READ1, CODDOP, BITRAT, BWPNAC, RFBW
  1 FORMAT (3F10.2, 1F15.2)
#2 READ2, THP1, THP2, THP3
#3 READ2, TS1, TS2, TS3
  2 FORMAT (3F10.6)
```

```

#4 READ3, NOEST, NJPCYL, IOPT9, TINC9, NGHCL, NLKCL
  3 FORMAT (3I5, 1F10.5, 2I5)
#5 READ4, DLLZET, DLLBLL, CODDOP, CNO, DLLBIT, ICOMM
  4 FORMAT (4F10.5, 1F10.2, 1I5)
#6 READ5, ATTN, TS1, TS2, TS3, BWPAC, IOPT
  5 FORMAT (4F10.5, 1F10.2, 1I5)
#7 READ6, IOPT
  6 FORMAT (1I5)

```

3.2.2.2 Output Parameters

After each successful sync attempt, statistics on the number of false entries into the lock and check modes and number of trials to sync are outputted. In addition, the following line of print is typed each successful trial:

```
PRINT 5, SUMTIM, TIME, ALTOTE, NLTOTE, SUMSWP, SUMSTP
```

In the above SUMTIM is the accumulated time to sync, TIME is the time to sync on previous attempt, ALTOTE is the number of successful sync attempts, NLTOTE is the total number code cycles searched SUMSWP is the average time to sync (T_{acq}), and SUMSTP is the average number of code cycles required to sync ($1/p$).

3.2.2.3 Options

The seven option parameters discussed above allow for a wide range of PN despreaders performance studies. Subsystem behavior as functions of noise estimate, loop operation, single or multiple hits per epoch can be easily evaluated via data card changes.

3.3 Performance

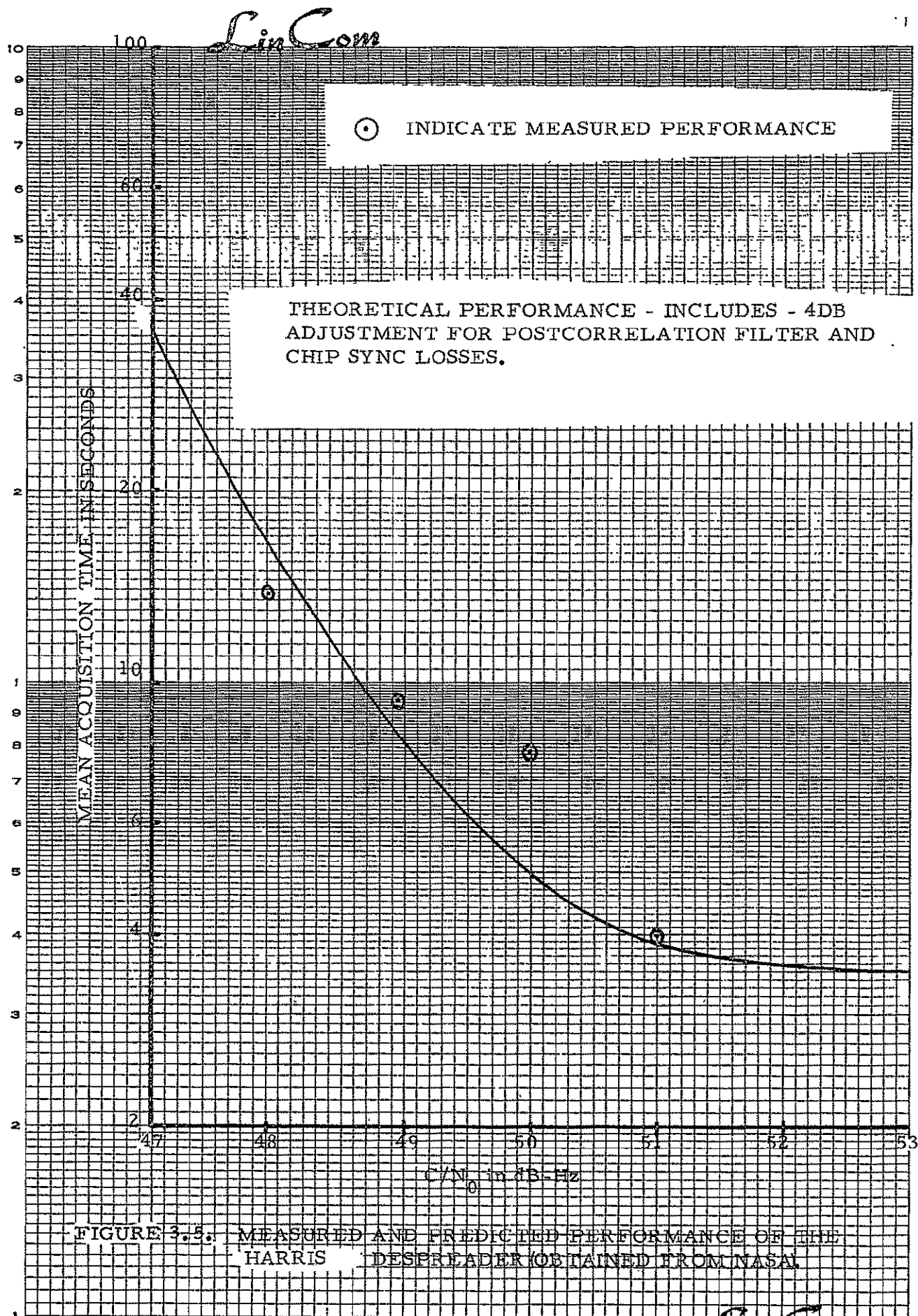
As previously stated, software described in this section is intended for use by NASA personnel in supporting laboratory testing of the Harris hardware despreaders. Prior to completion of software

ACQUISITION STRATEGY

SEARCH	$P_{FA} = 0.12$ $T_{INT} = 1 \text{ millisecond}$ Number of Stages = 1
CHECK	$P_{FA} = 0.0045$ $T_{INT} = 4 \text{ milliseconds}$ Number of Stages = 1
LOCK	$P_{FA} = 0.5$ $T_{INT} = 4 \text{ milliseconds}$ Number of Stages = 3

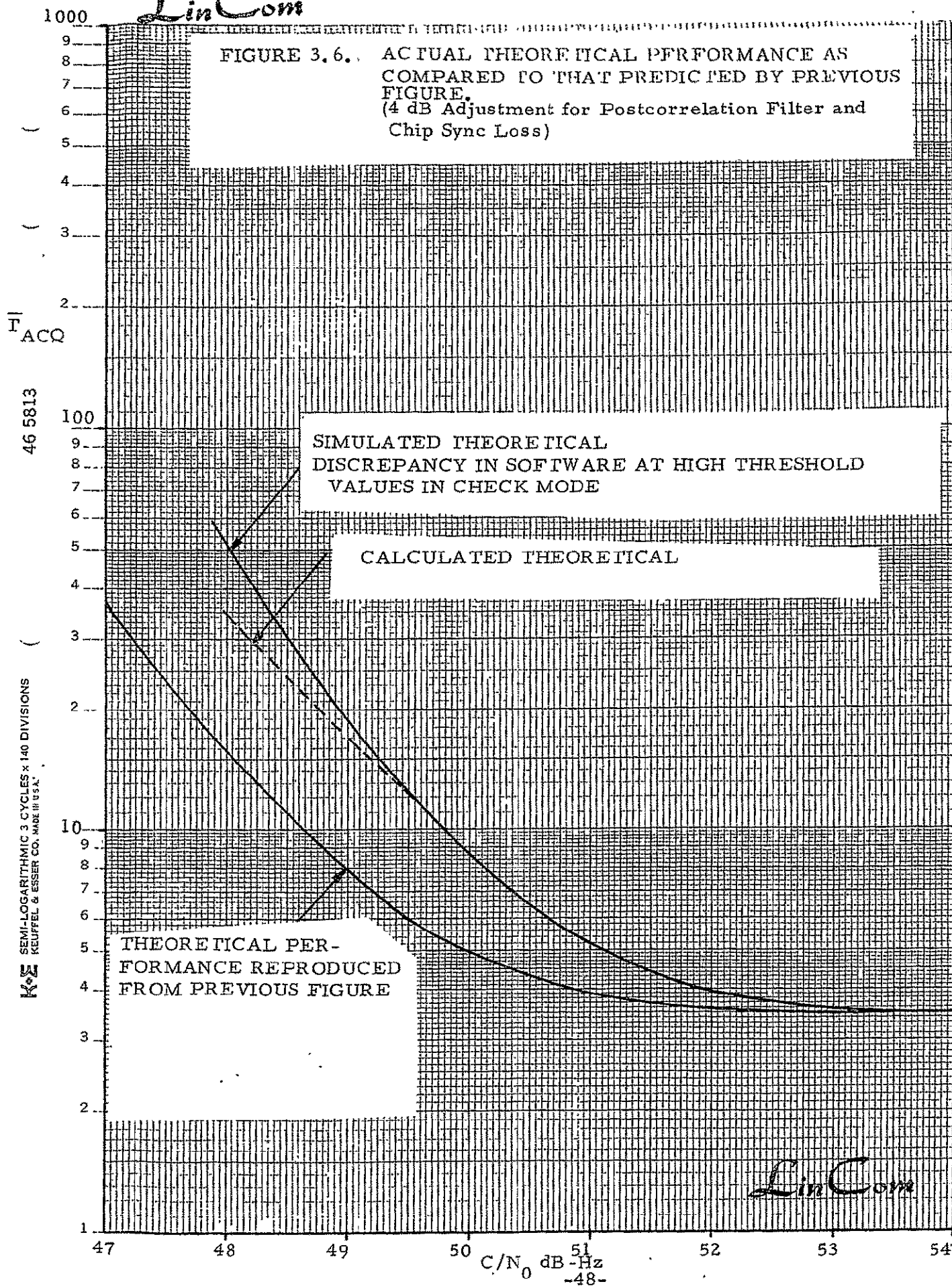
FIGURE 3.4 ESTL ALGORITHMS RECEIVED FROM NASA

PRECEDING PAGE BLANK NOT FILMED



LinCom

FIGURE 3.6. ACTUAL THEORETICAL PERFORMANCE AS COMPARED TO THAT PREDICTED BY PREVIOUS FIGURE.
(4 dB Adjustment for Postcorrelation Filter and Chip Sync Loss)



3.3.2 Performance Comparison

The actual simulated performance with the loops activated is shown in Figure 3.7. Also plotted are the measured data points reproduced from Figure 3.5 and the correct theoretical curve (loop behavior included as a flat 2.5 dB loss) of Figure 3.6. The results are in excellent agreement. It must be cautioned, however, that simulated data should be somewhat different from measured results because (1) simulation used 550 kHz IF filter bandwidths, (2) basic integration time parameter (K) was 29 not 32 with simulated results slightly scaled, (3) limited number of measurements made (at 48.3 dB C/N₀ hardware data based on 15 independent trials, software 25), (4) experimental accuracy not known and (5) ESTL data indicates some nonlinear behavior with signal to noise ratio. All of the above factors as well as some not listed reinforce and degrade measurement accuracy to some degree. As an example Figure 3.8 illustrates the variation in accuracy associated with (3) above.

LinCom has performed what may be termed a sort of "quick look" experiment. With the exception of possibly (5) above, the varying sources of errors can be carefully controlled once the software package is delivered to and exercised by NASA in direct support of testing. It is expected that excellent agreement will be found between simulated and measured data over a wide range of algorithms.

3.3.3 LinCom vs ESTL Algorithms - No Doppler

Figure 3.9 reproduced from page 93, Volume II of the Phase I report represents the performance of the LinCom algorithm. A comparison of the relative difference between the theoretical and actual (loop activated) performance curves of Figures 3.7 and 3.9 indicates that the ESTL algorithm using up/down counters is superior to LinCom's

LinCom

FIGURE 3.7. COMPARISON OF SIMULATED AND MEASURED RESULTS.

H_{ACQ}

46 5813

SEMI-LOGARITHMIC 3 CYCLES X 140 DIVISIONS
KEUFFEL & ESSER CO. MADE IN U.S.A.

THEORETICAL PERFORMANCE

ESTL MEASURED DATA

SIMULATED PERFORMANCE

LinCom

47 48 49 50 C/N_0 51 52 53 54
dB-Hz -50-

Figure 3.8

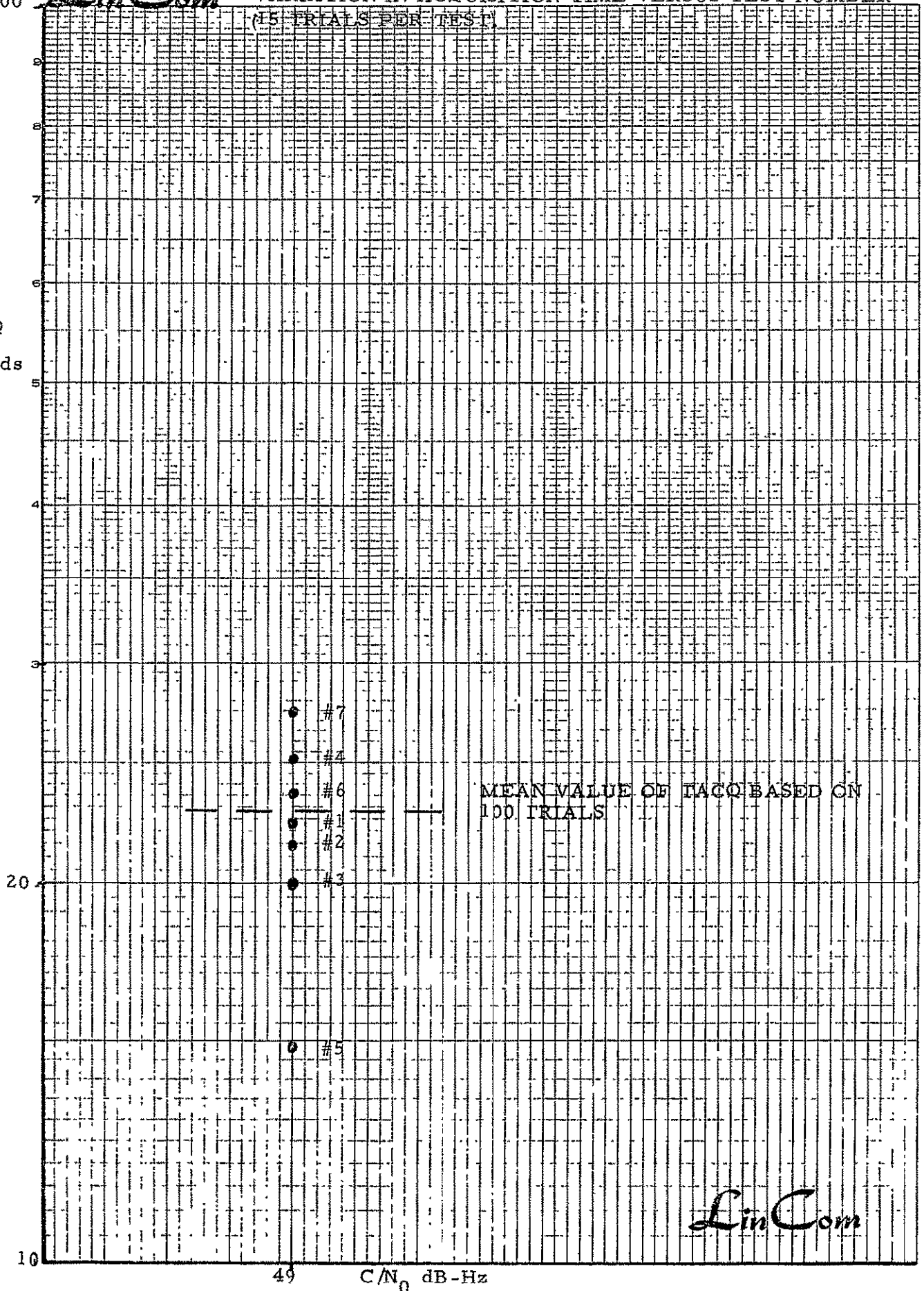
VARIATION IN ACQUISITION TIME VERSUS TEST NUMBER

(15 TRIALS PER TEST)

T_{ACQ}
Seconds

EUGENE DIETZEN CO.
MADE IN U. S. A.

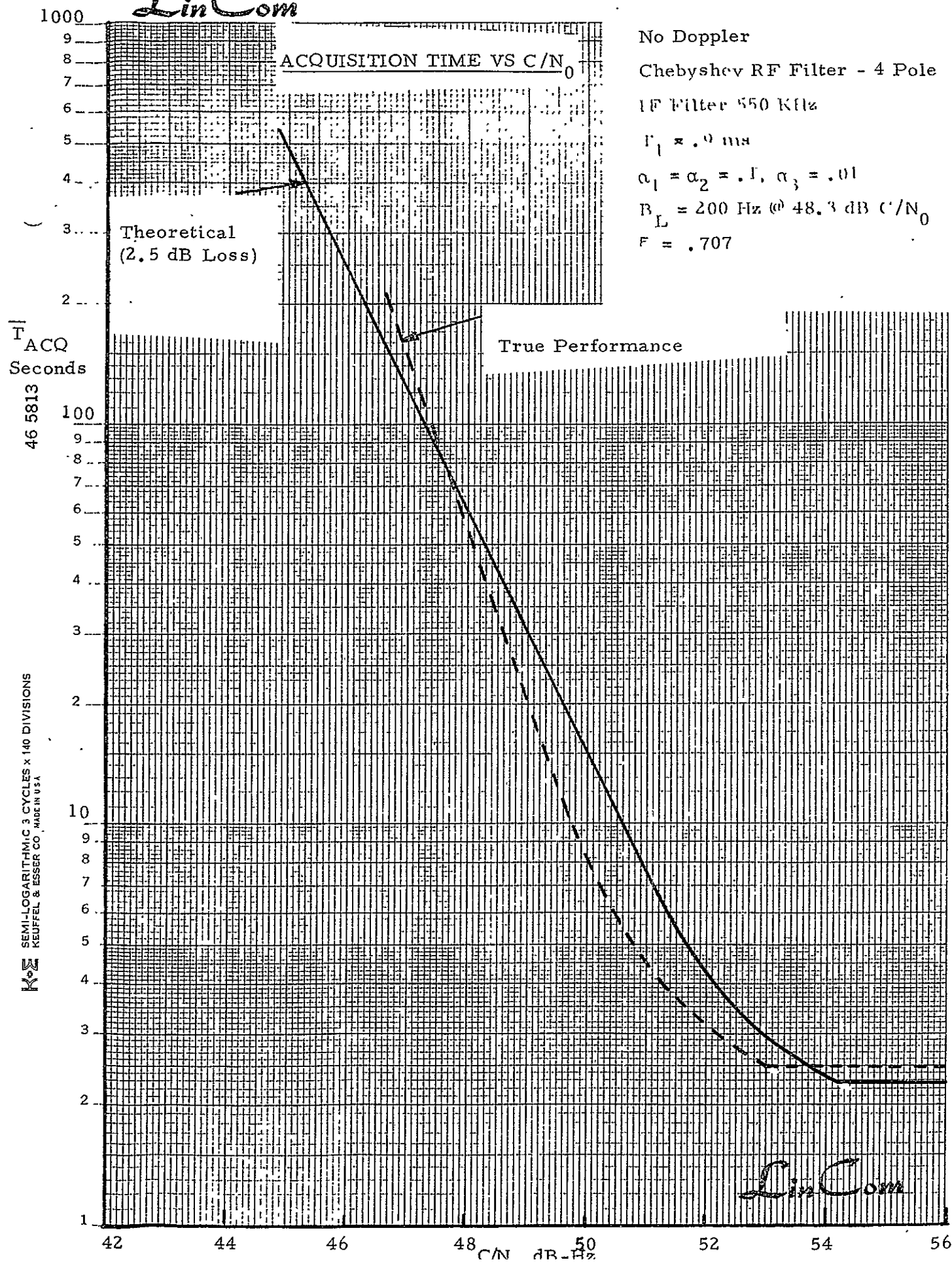
NO 340R-L110 DIETZEN GRAPH PAPER
SEMI-LOGARITHMIC
1 CYCLE X 10 DIVISIONS PER INCH



LinCom

LinCom

FIGURE 3.9 LINCOM PERFORMANCE.



use of reset counters when loop behavior is considered. It is also worth noting that any theoretical calculation which does not model loop behavior correctly will not predict true performance.

3.3.4 LinCom vs ESTL Algorithm - Doppler

At the time of this writing the Harris unit is not capable of operating with doppler due to a bandwidth switching problem, so that no measured data is available. The simulated performance with +300 code chips of doppler is shown in Figure 3.10, with extremely poor operation indicated. Only a limited number of samples were used in generating this curve as it is apparent that the algorithm and/or threshold settings must be changed. It is felt that when suitable algorithms or thresholds are selected the no doppler performance will be somewhat degraded.

The poor performance with doppler can be understood by comparing the theoretically predicted results shown in Figures 3.7 and 3.9 at a high value of C/N_0 . The LinCom algorithm has a mean time to acquire of about 2.35 seconds, corresponding to a mean time to dwell at each search position of,

$$\overline{T}_D = \frac{(2)(2.35)}{(2)(2047)} = 1.15 \text{ milliseconds};$$

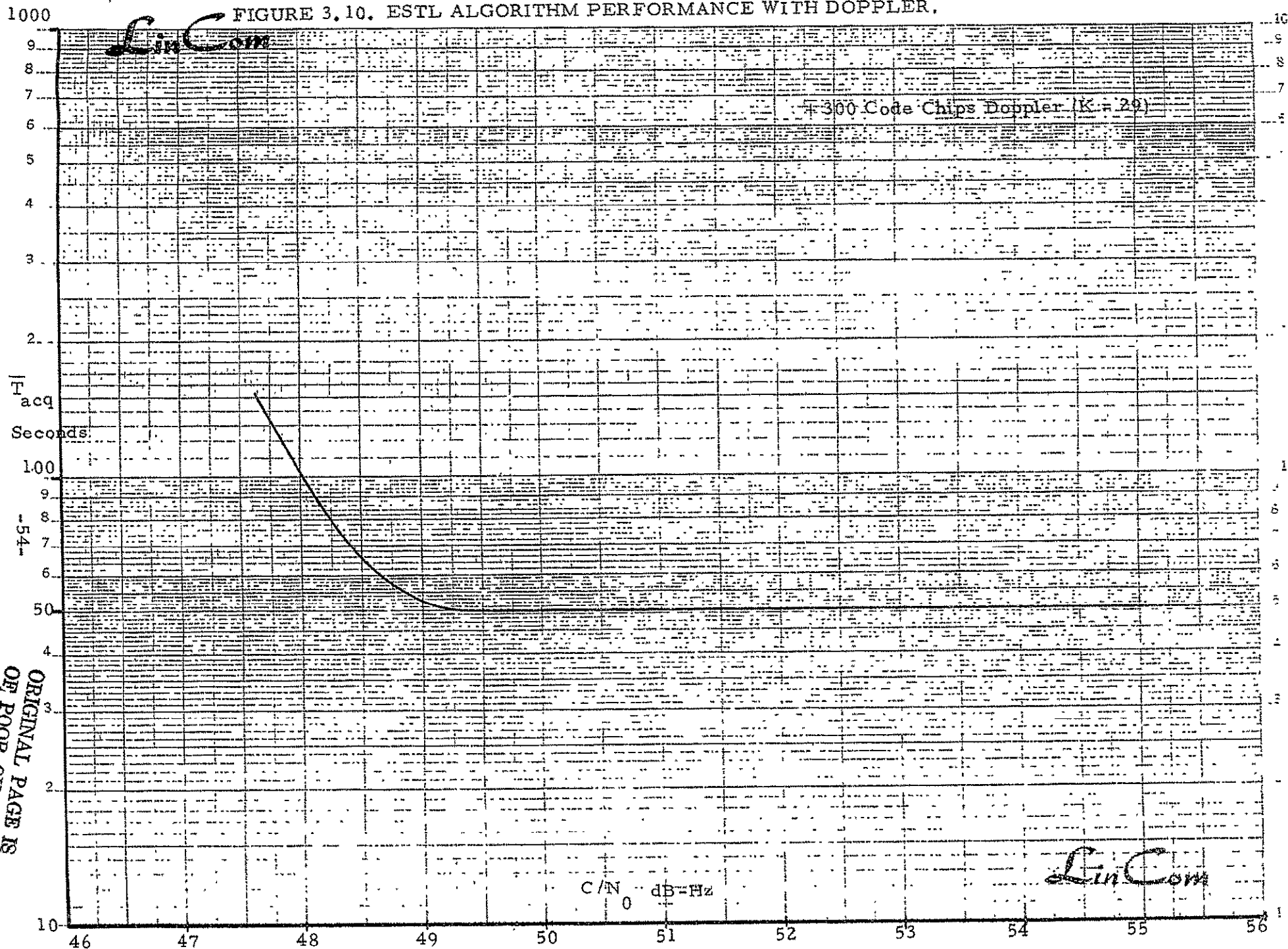
that of the ESTL algorithm is

$$\overline{T}_D = \frac{(2)(3.5)}{(2)(2047)} \times \frac{(.903 \times 10^{-3})}{(1. \times 10^{-3})} = 1.54 \text{ milliseconds.}$$

With Δf of doppler, effective dwell time is increased by

$$\overline{T}_{DE} = \frac{\overline{T}_D}{1 - 2\Delta f \overline{T}_D}.$$

FIGURE 3.10. ESTL ALGORITHM PERFORMANCE WITH DOPPLER.



ORIGINAL PAGE IS
OF POOR QUALITY

A value of $\overline{T}_D$ of 1.54 milliseconds is much too high and must be reduced by increasing the thresholds (reduce false alarms) or by adding more stages to the search state.

The effects of loop behavior with and without doppler and the multiple hits available per epoch with positive doppler can be seen from the data displayed in Table 3.1. The following points are notable:

1. A comparison of N for 0 and -300 Hz doppler indicates the degradation associated with loop behavior under stress.
2. A comparison of N for +300 and -300 Hz of doppler illustrates the marked improvement due to the multiple changes to synchronize with positive doppler.

TABLE 3.1. MEAN NUMBER OF CODE CYCLES SEARCHED
TO SUCCESSFULLY SYNCHRONIZE AT A
 C/N_0 of 48 dB-Hz - ESTL ALGORITHM.

<u>DOPPLER</u>	<u>N*</u>
0	3.4
+300	1.5
-300	13.7

*10 Trials per Measurement

SECTION 4

SYMBOL SYNCHRONIZER SUBSYSTEM

4.0 Introduction

A somewhat idealized temporary symbol synchronizer, not capable of operating in deep noise, was included as part of the Carrier Recovery Subsystem during the Phase I effort. Its purpose was to evaluate the effects that carrier recovery hardware and PN despreaders equipment has on symbol error rate performance. A Symbol Synchronizer Subsystem (SSS) capable of tracking the Manchester encoded symbols in deep noise has been developed to replace the previous temporary design. This subsystem is simulated in Subroutine SSS. Section 4.1 describes the trade-offs that went into selecting the hardware implementation. A description of how this implementation achieves the symbol synchronization and data detection functions will be found in section 4.2. The analyses associated with predicting theoretical performance is documented in section 4.3. Section 4.4 contains a description of a series of experiments used to verify and validate the simulation. The developed software is discussed and documented in section 4.5.

4.1 Symbol Synchronization Techniques and Tradeoffs

Our selected, approach for the hardware simulation is an all-digital implementation incorporating a digital data transition tracking loop, Reference 1. This section presents the trade-offs which led to this approach. The following sections discuss the inherent problems and approaches to providing symbol synchronization for coded communication systems which must give near optimum performance at low SNR's.

The classes of symbol synchronization systems covered in the discussions are:

- Harmonic generating/tracking symbol synchronizers
- MAP symbol synchronizers
- A class of early-late gate type symbol synchronizers (absolute value, least of squares, I-Q loops)
- I-Q loop symbol synchronizers
- Digital data transition tracking loop synchronizers
- Symbol sync systems operate in two distinct modes: the signal or sync acquisition mode and the synchronous or tracking mode (Reference 2, Chapter 10). The signal acquisition mode relates to system performance during the time the clock signal is being established, while the tracking mode relates to system performance as data detection is being accomplished. Each mode has fundamental physical restrictions and characteristics. The best performance is achieved when these two modes of operation are as independent as possible. Performance indices are different for the two modes. Those considered in the following discussion are:

<u>Sync Acquisition Mode</u>	<u>Synchronous or Tracking</u>
•Sync acquisition range	•Symbol sync jitter
•Sync acquisition time	•Symbol slippage rate
•Probability of sync acquisition	•Symbol error probability degradation
•Cost/Complexity of hardware	•Static phase error
	•Cost/Complexity of hardware

Symbol synchronizers can be implemented with either analog or digital circuits. The performance of each has been compared and it can be shown to be approximately equal, with the digital implementation exhibiting only a very small degradation in accuracy, Ref. 4. The advantages of digital mechanization are stability, accuracy, and reliability.

4.1.1 Harmonic Generating-Tracking Symbol Synchronizers

Perhaps the oldest approach to the problem of producing symbol synchronization from the information bearing signal is illustrated in Figure 4-1. Since symbol sync information is

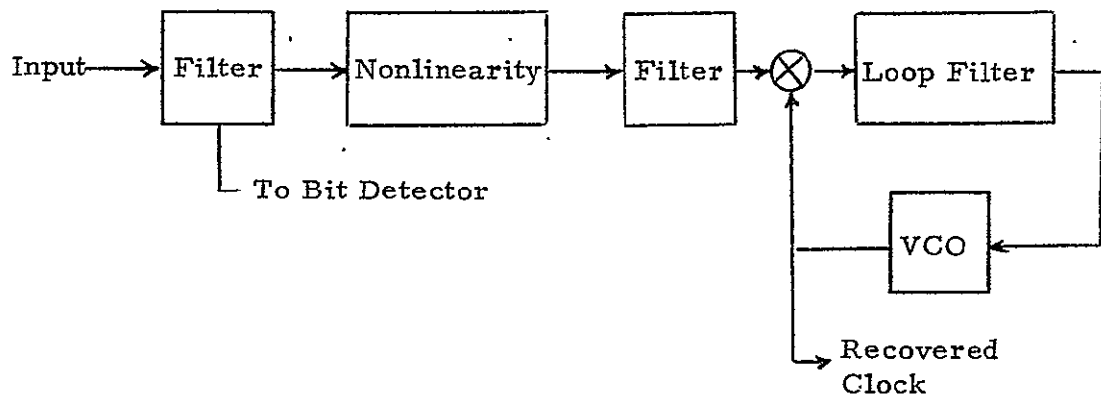


Figure 4-1. Simple Rectifier-Filter Type Symbol Synchronizer.

carried in the data transitions, a nonlinear operation (e.g., a differentiating circuit followed by a rectifier) can be used to generate a clock component whose strength depends upon the data transition density, the signal data power, and the bit rate. For Manchester encoded data, the ratio of the power in the clock component to the total data power is given by

$$\frac{P_c}{P_{\text{total}}} = 31[1-p(1-p)]^2 \hat{R}^2$$

where p is probability of occurrence of a "1" in the data stream and $\hat{R}$ is the bit rate. The unfortunate feature of such a system is that the differentiation of a noisy signal produces a noisier signal and also generates a sizable self-jamming/self-noise component

whose power is also proportional to the bit rate squared. For high input SNR's and fixed data rate signals, such as high-rate QPSK where $E_b/N_0 = 5$ to 10 dB and the data rate is fixed, the thermal effects and the self-noise generated can be simultaneously tolerated. However, at low input SNR's, $E_b/N_0 < 3$ dB, the self-noise term and thermal noise term are of the same order and marked degradation (6 dB) in system performance results. Thus at high noise levels relative to that of the data signal, the data transitions are erroneously generated, and the phase-locked oscillator tends to follow the noise rather than the clock component.

An alternate harmonic tracking symbol synchronizer is illustrated in Figure 4-2. The noise performance of this type of symbol

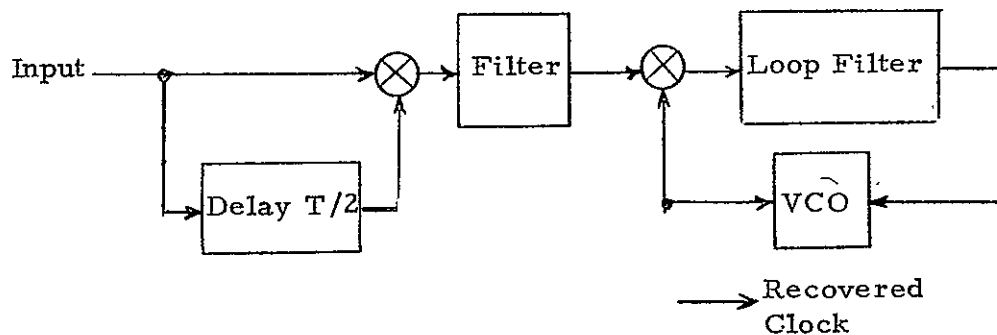


Figure 4-2. Symbol Synchronizer for Digital Data and High SNR.

synchronizer is known to be superior to that of Figure 4-1 as the operation does not depend on the data transitions of the signal component directly. The synchronizer does, however, exhibit a sharp performance threshold for low input SNR's due to the product of the noise with its delayed version. Superior noise performance

can be obtained from the symbol synchronizers discussed below, and no further reference will be made to this class.

4.1.2 Symbol Synchronizer Based Upon MAP Estimation Theory

The process of extracting the necessary timing from the information-bearing waveform for coherent detection of the digital stream has been referred to as "data derived sync" or "self-sync". The problem of estimating symbol sync from a received signal-plus-noise can be optimally handled using the theory of maximum likelihood estimation for an unknown parameter in Gaussian noise (Reference 3, Chapter 9). In this theory one assumes complete knowledge of an observed sequence of K data bits and proceeds to arrive at the maximum likelihood estimate of bit sync. A practical interpretation of this in the time domain suggests correlating the received signal with a stored replica of the basic received pulse shape and taking the log hyperbolic cosine of this result and accumulating these values over all possible bit sync possibilities. The value which yields the largest accumulated value is then declared the best symbol sync estimate.

Unfortunately, this approach to symbol sync suffers from two difficulties. First, the values for the symbol sync epoch take on a continuum of values over the bit duration. To practically apply this theory to the development of a practical PCM bit sync system, one must quantize the interval $[0, T]$ into a number of levels and perform a parallel search procedure over these allowable bit sync positions. Needless to say, a parallel search is prohibitive of equipment while a serial search is prohibitive in terms of bit sync acquisition time. The second disadvantage of applying this theory to the development of a practical symbol sync system has to do with the

fact that it is open loop, i.e., no provision is made in the theory for operation in the tracking mode. The symbol sync jitter performance of the optimum maximum likelihood estimate synchronizer in terms of the ratio of energy per bit to noise spectral density (ST/N_0) for various values for K and various pulse waveform shapes is given in Reference 3, Chapter 9, pp. 420-429.

To include the tracking function into the requirements of a symbol sync system, various authors (Reference 3, Chapter 9) propose tracking the symbol sync with a closed loop system whose error signal, generated from the open loop maximum likelihood estimate, is used to control a voltage control oscillator (VCO). We now discuss a large class of symbol sync systems which are based on estimation theory.

The functional diagram of a closed loop symbol synchronizer based on MAP estimation theory is illustrated in Figure 4-3. The phase of the timing pulse generator, which controls the start and termination of the integrate-and-dump circuits, is bumped every T sec by an amount proportional to the magnitude of the gradient of the a posteriori estimate of the bit sync epoch ϵ and in a direction based on its sign as computed for the previous KT sec. Notice the nonlinearity $\tanh x$ in the in-phase channel. We note for large SNR's that $\tanh x = \text{sgn } x$, while for small SNR's, $\tanh x = x$. For high SNR's the synchronizer is reminiscent of the classical I-Q bit synchronizer.

Other symbol synchronizers based on MAP estimation theory are of the early-late gate type. The functional diagram for an early-late gate type symbol synchronizer which destroys the data modulation by taking absolute values to generate a tracking S-curve is illustrated in Figure 4-4. In Figure 4-5 an early-late gate symbol synchronizer with square-law type of nonlinearity is illustrated. The detailed analysis of performance can be found in Reference 3, Chapter 9.

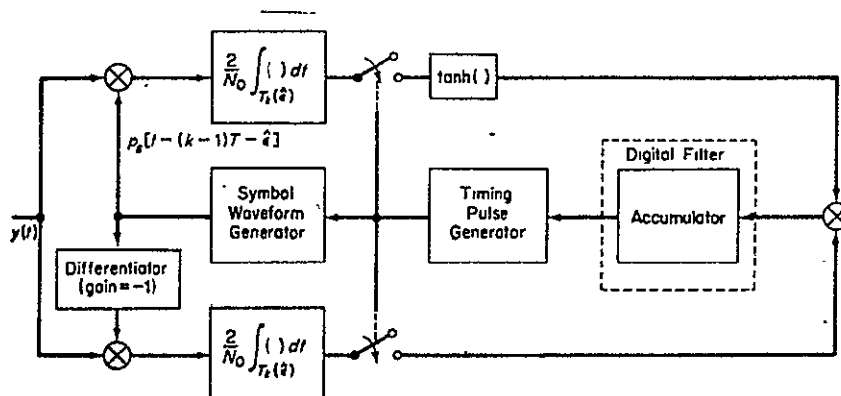


Figure 4-3. A Closed Loop Symbol Synchronizer Motivated by the MAP Estimation Approach (Taken from Lindsey-Simon, Chapter 9, Reference 3).

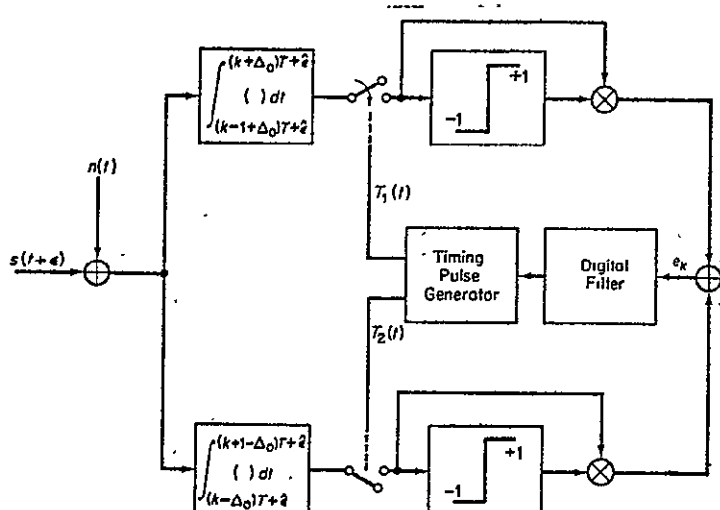


Figure 4-4. Functional Block Diagram of an Early-Late Gate Type of Symbol Synchronizer with Absolute Value Type of Nonlinearity (Taken from Lindsey-Simon, Chapter 9, Reference 3).

4.1.3 Symbol Synchronizers Motivated by Optimum Design of the Phase Detector

There are several approaches to the problem of selecting an optimum phase detector characteristic. What is optimum depends on the performance measure chosen to represent system behavior. For example, during the signal acquisition mode, the performance

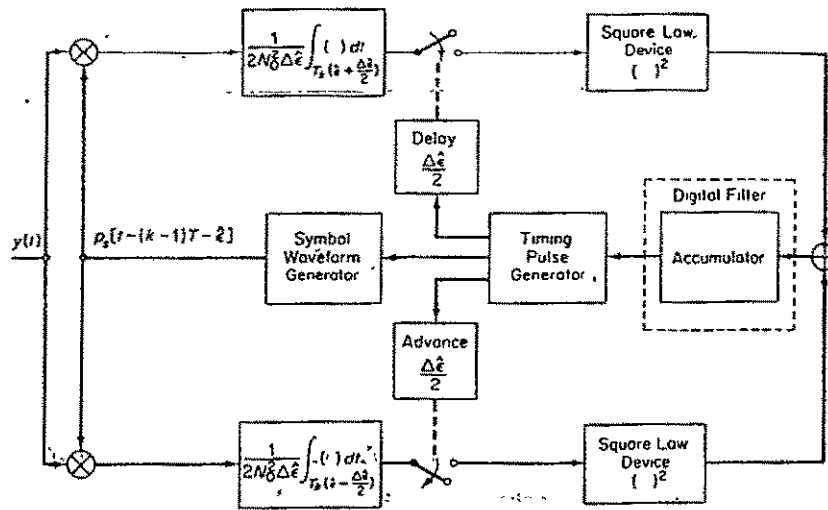


Figure 4-5. An Early-Late Gate Type of Bit Synchronizer with Square-Law Type of Nonlinearity (Taken from Lindsey-Simon, Chapter 9, Reference 3).

measures are acquisition time and range, and probability of acquisition. After symbol sync has been acquired, attention is focused on the tracking mode where measures such as mean squared bit sync jitter, mean time to first loss of sync, and bit slip rate become significant. In the past, several approaches have been addressed:

- Acquisition mode

- Choice of the phase detector characteristic to maximize acquisition range (Reference 2, Chapter 10).
- Choice of the phase detector characteristic to minimize acquisition time (Reference 2, Chapters 10 and 11).

- Tracking Mode

- Minimization of the area under the tail of the symbol sync error probability density function (Reference 1, Chapter 9; Reference 1, Chapter 11).
- Minimization of the k^{th} absolute central moment of the symbol sync error pdf (Reference 1, Chapter 10 and 11).

4.1.4 Digital-Data Transition Tracking Symbol Synchronizers

A symbol sync system originally proposed by Lindsey and Tausworthe, Reference 1 (and now operational in the Deep Space Network and elsewhere) for demodulation of coded telemetry signals is the digital data transition tracking loop illustrated in Figure 4-6. This was based on an optimum design of the phase detector in the tracking mode.

In this design, the input noise-free signal $s(t, \epsilon)$ is a random pulse train characterized specifically as the rectangular pulse defined by $p_s(t) = \sqrt{S}$ for $0 \leq t \leq T$, $p_s(t) = 0$ for all other t . The sum of this signal plus noise $n(t)$ is passed through two parallel branches, which are triggered by a timing pulse generator according to a digitally filtered version of an error signal formed from the product

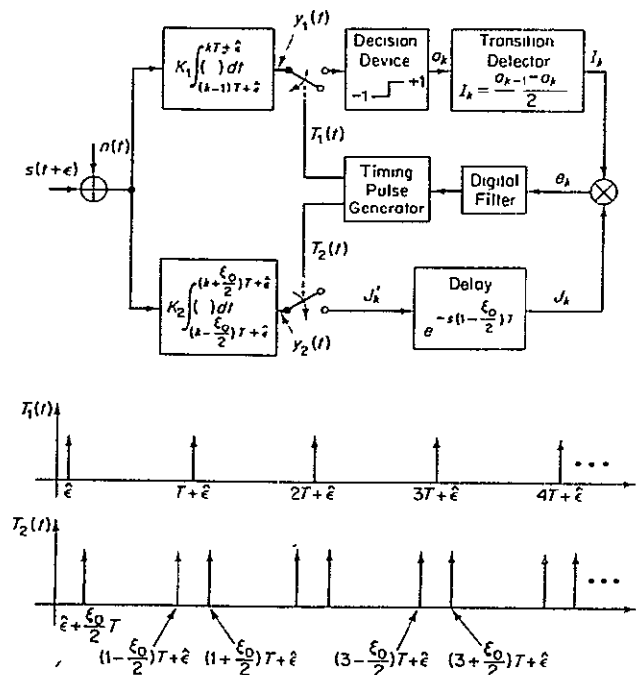


Figure 4-6. The Digital Data Transition Loop (DTTL) (taken from Lindsey-Simon, Chapter 9, Reference 3).

of the branch outputs. Furthermore, two branches are held at a fixed phase relationship with one another by the timing generator. Basically, the in-phase branch monitors the polarity of the actual transitions of the input data and the mid-phase branch obtains a measure of the lack of sync. The particular way in which these two pieces of information are derived and combined to synchronize the loop is described in detail in Reference 3, Chapter 9, pp. 422-458.

While the configuration in Figure 4-6 is configured for operation on NRZ data it can be reconfigured for Manchester by operating at twice the bit rate, i.e., replace T by T/2 in Figure 4-6.

4.1.5 Performance Comparison for Various Symbol Synchronizers

4.1.5.1 Tracking Mode/RMS Symbol Sync Jitter

In comparing the performance of several different bit sync configurations, one must choose a fixed operating condition that is common to all. For the comparison we set the SNR $R_d = E_b/N_0 = -5$ dB (-2 dB in a one-half bit rate bandwidth), i.e., the Shuttle threshold design point, and considered the normalized rms bit sync jitter $\sigma_\lambda\%$ as the performance measure. Since it is known (Reference 3, Chapter 9) that the difference of square loop performance is inferior to the absolute value type synchronizer as well as the digital data transition loop synchronizer, the performance comparison is only made here between these latter two. Furthermore, it has been shown (Reference 3, Chapter 9) that the optimum setting of the earlier integrator relative to the late integrator in the absolute value synchronizer is $\Delta_0 \simeq \frac{1}{4}$.

Table 4-1 compares the rms jitter performance at the Shuttle brassboard design point of the digital data transition synchronizer to that of the absolute value synchronizer for various normalized

Table 4-1. Comparison of the Absolute Value Synchronizer to DTTL Synchronizer for Two Different Window Widths ξ_0 .

NRZ-L		AV, $\Delta_0 = 1/4$	DTTL; $\xi_0 = 1/2$	DTTL; $\xi_0 = 1/4$		AV ($\Delta_0 = 1/4$) DTTL ($\xi_0 = 1/4$)	AV ($\Delta_0 = 1/4$) DTTL ($\xi_0 = 1/2$)
$R_d = E/N_0$	$B_L T$	$\sigma_{\lambda} \%$	$\sigma_{\lambda} \%$	$\sigma_{\lambda} \%$	$360 \sigma_{\lambda}$, degrees		
-5dB	1.0	19.8655	17.586	11.94	43	2.2 dB	0.53 dB
-5dB	0.50	14.0470	12.435	8.44	30	2.2 dB	0.53 dB
-5dB	0.25	9.9327	8.79	5.97	22	2.2 dB	0.53 dB
-5dB	0.05	4.4420	3.93	2.67	9.6	2.2 dB	0.53 dB
-5dB	0.0167	2.5646	2.27	1.54	5.5	2.2 dB	0.53 dB
-5dB	0.0143	2.3743	2.10	1.43	5.14	2.2 dB	0.53 dB
-5dB	0.011	2.0940	1.85	1.25	4.5	2.2 dB	0.53 dB
-5dB	0.010	1.7120	1.53	1.03	3.7	2.2 dB	0.53 dB

AV = Absolute value synchronizer

DTTL = Digital data transition loop synchronizer

loop bandwidths; that is, the loop bandwidth times the bit duration $B_L T$; and two different window widths, ξ_0 in the digital data transition loop synchronizer. It is significant that:

- The absolute value synchronizer rms bit sync jitter is 2.2 dB inferior to that of the digital data transition loop synchronizer with $\Delta_0 = \xi_0 = \frac{1}{4}$ for $R_d = -5$ dB.
- The absolute value synchronizer loop rms bit sync jitter is 0.53 dB inferior to that of the digital data transition loop synchronizer with $\Delta_0 = \frac{1}{4}$, $\xi_0 = \frac{1}{2}$ for $R_d = -5$ dB.

Therefore, on the basis of this performance measure the digital data transition tracking loop synchronizer is the recommended mechanism and was selected for the hardware simulation.

4.1.6 Acquisition Performance Comparisons for the DTTL and Absolute Value Type Symbol Synchronizers

This section compares the acquisition performance of DTTL and absolute value synchronizers. Since the theory of signal

acquisition in the presence of noise is not complete, we present here the key results obtained by computer simulation. Here acquisition time was determined as a local clock frequency. The time to acquire was defined as the time to stop slipping cycles plus the time for the loop to settle in phase to 5% of a bit period. Figure 4-7 plots the acquisition performance of the two bit synchronizers under consideration for three different input conditions. These are with a noise free square wave input, with a noise free NRZ input, and with noisy random NRZ with an $E_b/N_0 = 2.5$ dB. In addition,

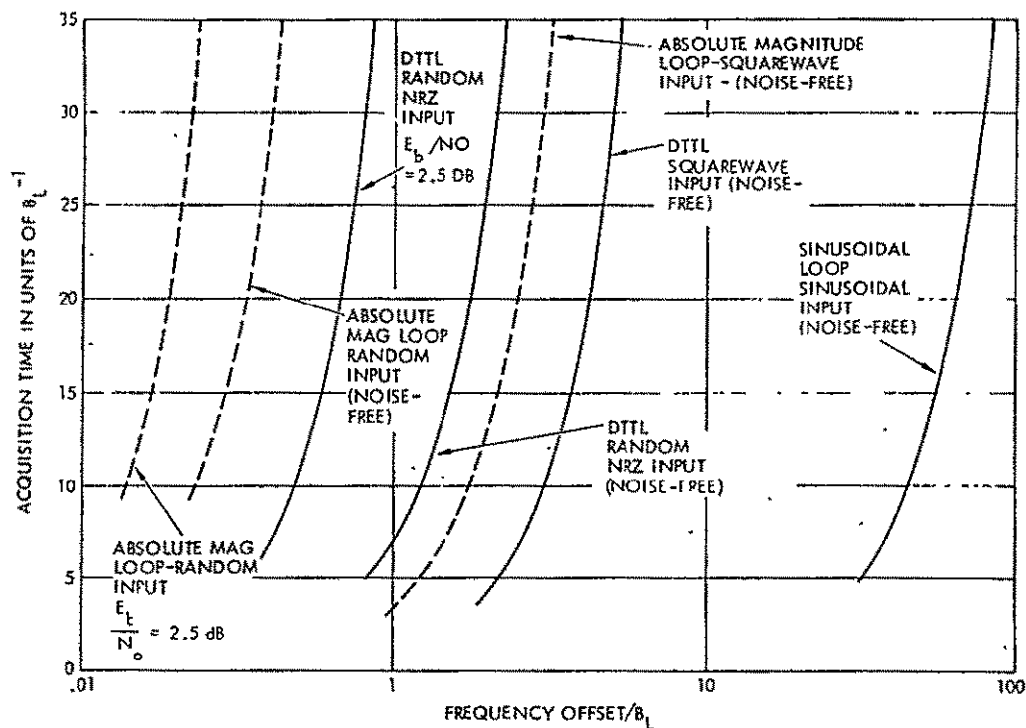


Figure 4-7. Acquisition Time versus Frequency Offset to Bandwidth Ratio (taken from MacRae-Smith,

the acquisition performance of a continuous wave PLL as taken from Lindsey (Reference 2) is shown. Several conclusions should be noted:

- The amount of frequency offset that can be tolerated is much less than one would predict using the standard sinusoidal PLL theory.
- At low SNR's, acquisition times can be excessive and the acquisition range is considerably degraded.
- The acquisition time, for a given input frequency offset, is strongly dependent upon whether the input data is periodic, random, noise-free, or noisy.
- The DTTL loop is superior to the early-late gate absolute value synchronizer under all input conditions.
- With random NRZ, the DTTL can tolerate nearly four times more frequency uncertainty than that of the absolute value synchronizer.
- The acquisition time is highly dependent on the transition density in the data stream.

On the basis of these results, the digital data transition tracking loop synchronizer is again the recommended approach for hardware simulation as it is capable of outperforming the absolute value type in both the acquisition mode and tracking mode.

4.1.7 Bit Error Probability Degradation Due to Symbol Sync Jitter

Because the effects of bit sync jitter on the bit error probability remain an unsolved theoretical problem in systems which employ convolutional codes, we present here the effects of bit sync jitter on the performance of uncoded coherent communications which use Manchester coding. These curves, illustrated in Figure 4-8 represent plots of the bit error probability versus $R_d = ST/N_0$ for various values of the normalized rms bit sync jitter σ_λ .

4.1.8 Summary of Selected Symbol Synchronizer Configurations

Based on the foregoing discussions, symbol synchronization in the presence of frequency offset and low SNR's is best accomplished

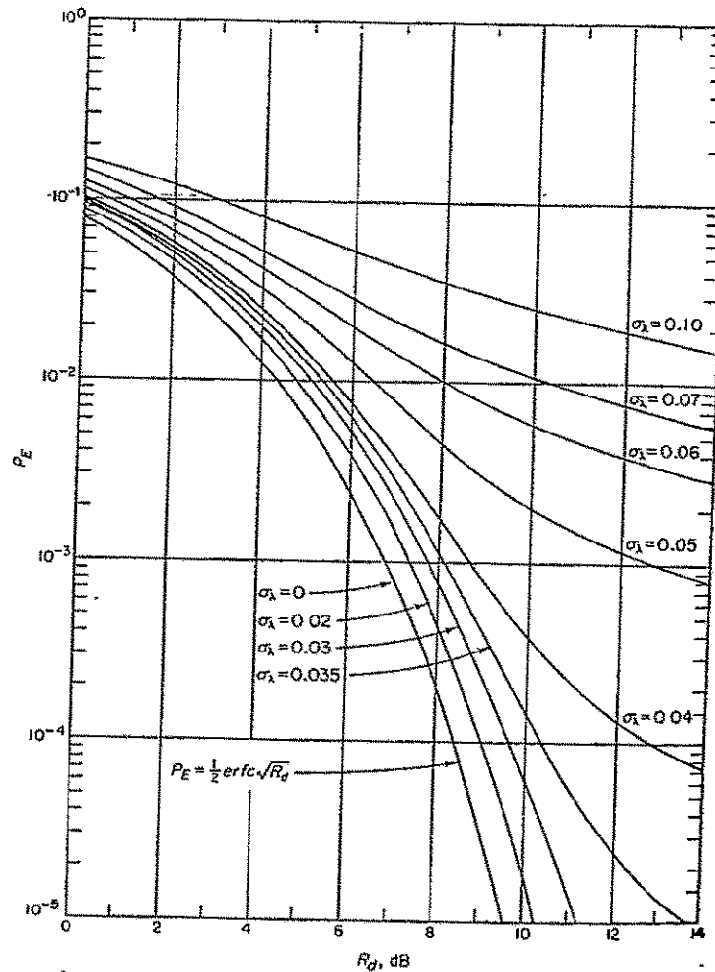


Figure 4-8. BER versus SNR for Various Values of Jitter.

by the window-optimized digital data transition tracking loop synchronizer. This synchronizer outperforms a class of early-late gate type symbol synchronizers which is illustrated in Table 4-2. Based upon the tradeoffs given in this section LinCom has selected the DTTL for hardware simulation.

4.2 Symbol Synchronizer Operation

A block diagram of the Symbol Synchronizer Subsystem selected for simulation is shown in Figure 4.9 . The demodulated symbol stream from the Costas loops lower arm mixer is processed by inphase and quadrature arm hardware. An in phase arm integrator is used to integrate over half symbol intervals under the control of timing generated by a local VCO. Data is detected at the end of each symbol by subtracting the integrated value over the second half symbol from the value (stored) of the integration over the first half symbol.

A quadrature arm integrator is used to integrate the incoming signal over a relatively narrow time interval straddling all possible symbol transition points. It is this arm which derives a signal whose magnitude is a function of the time difference between the incoming symbol stream and the local VCO. Figure 4.10 will be used to illustrate this point. A single Manchester symbol is shown as a function of time (measured with respect to transmitted signal). Below this curve is a plot of window interval over which the quadrature arm will integrate this signal (time measured with respect to local VCO). The bottom curve represents the integrator output voltage at the end of the window interval as a function of time differential. This voltage is zero when the local VCO aligns the window to perfectly straddle the mid-transition of the incoming symbol (shown as condition N) since the area under curve represents equal amounts of negative and positive signal. A positive voltage is produced when our local VCO slews to the right (condition #1) and a negative voltage results from a slew to the left (condition #2). Hence the output voltage is a measure of the relative time difference between the incident symbol and local VCO timing.

In a conventional loop this output voltage, after low pass filtering, would be used to synchronize our VCO with the incoming symbol

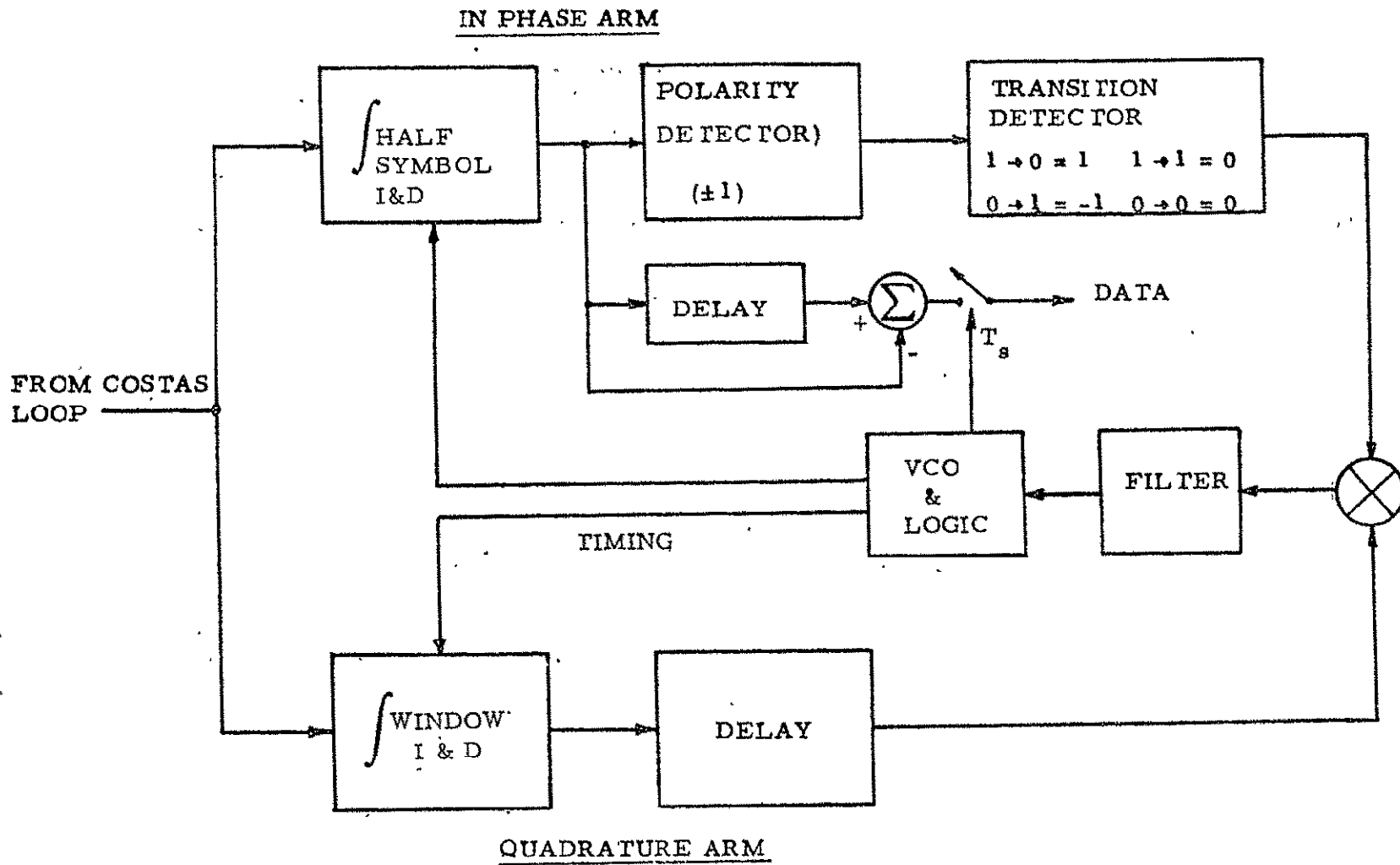


FIGURE 4.9. SYMBOL SYNCHRONIZER BLOCK DIAGRAM

LinCom

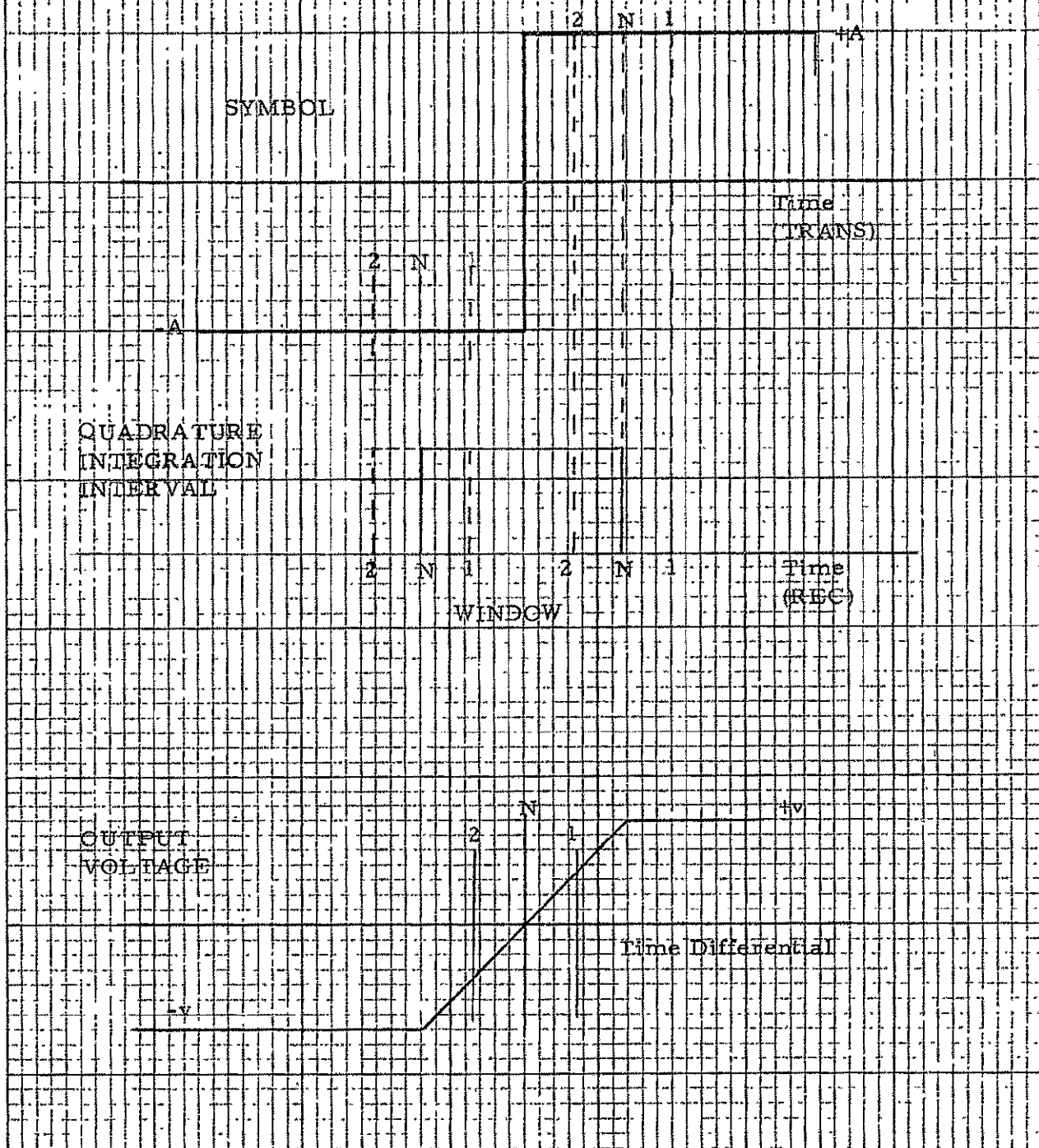


FIGURE 4.10. QUADRATURE ARM SETS PHASE DETECTOR CHARACTERISTIC.

LinCom

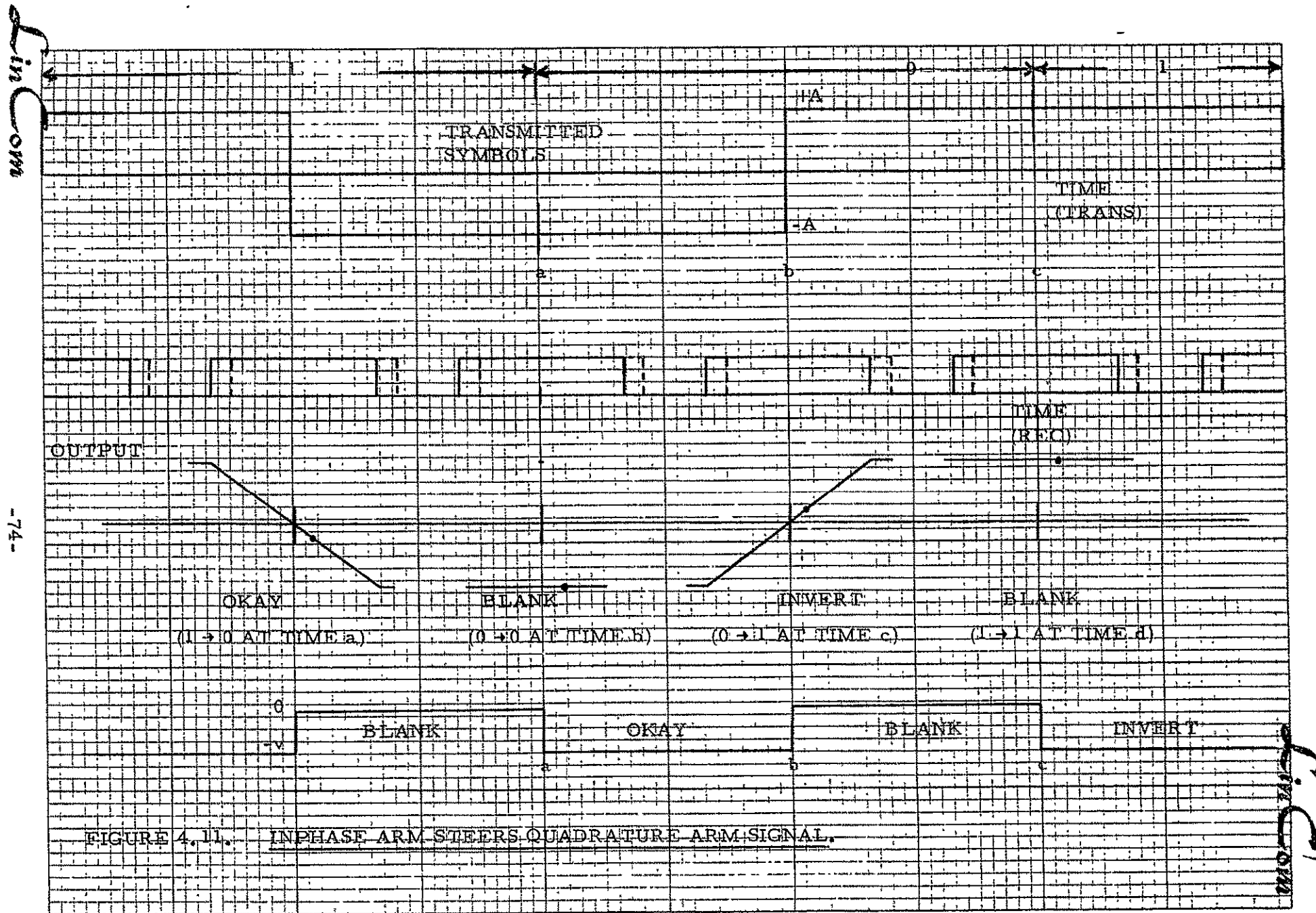


FIGURE 4.11. INPHASE ARM STEERS QUADRATURE ARM SIGNAL.

timing (drive error signal to zero). Unfortunately the quadrature arm itself is not capable of performing this task as the polarity of the error signal is sensitive to the half symbol transition polarity. This fact will be illustrated with the aid of Figure 4.11. The top curve depicts a 101 symbol pattern. Directly below this curve is shown the quadrature arm integration window. Note that integration takes place twice each symbol (at every possible transition point). The third curve shows the integrator output voltage as a function of time difference at each possible transition point. The window straddling the "0" symbol mid-transition (point b) corresponds to the condition of Figure 4.10. With the VCO timing slewed to the right (broken lines) the integrator output is positive with a value shown as the dot on the third curve. For the mid-transition integration of the "1" symbol the output is zero when perfectly aligned but is negative for a VCO slew to the right. The output voltage corresponding to the window intervals straddling the start of symbol transition such as points (a, c) are insensitive to the VCO slew and take on a large negative or positive value depending upon the adjacent half symbols (0 $\rightarrow$ 0 negative, 1 $\rightarrow$ 1 positive). One can see that a low pass filtering of these sample values will produce zero drive voltage for the VCO (no pulling force) independent of the time difference.

Using information about the half symbol transition direction alleviates this problem. The rule is quite simple. If a 1 $\rightarrow$ 0 transition occurs the output of the quadrature arm integrator is passed directly into the low pass loop filter. If a 0 $\rightarrow$ 1 transition occurs the output voltage is inverted before being passed into the loop filter. If no transition occurs (0 $\rightarrow$ 0, or 1 $\rightarrow$ 1) the output is prohibited from driving the loop filter. The net effect is to produce, for a VCO slew to the right, a negative error signal proportional to the time difference

at each half symbol transition and zero signal when no transition occurs. The lower waveform of Figure 4.11 illustrates this point. A positive going waveform would result if the VCO was slewed to the left. Note that the output samples are delayed and are not applied to the loop filter until the end of the half symbol. As an example, the sample of the integration over the mid-transition at point b is not passed to the loop filter until the time starting at point c. This delay is inherent in evaluating the transition direction.

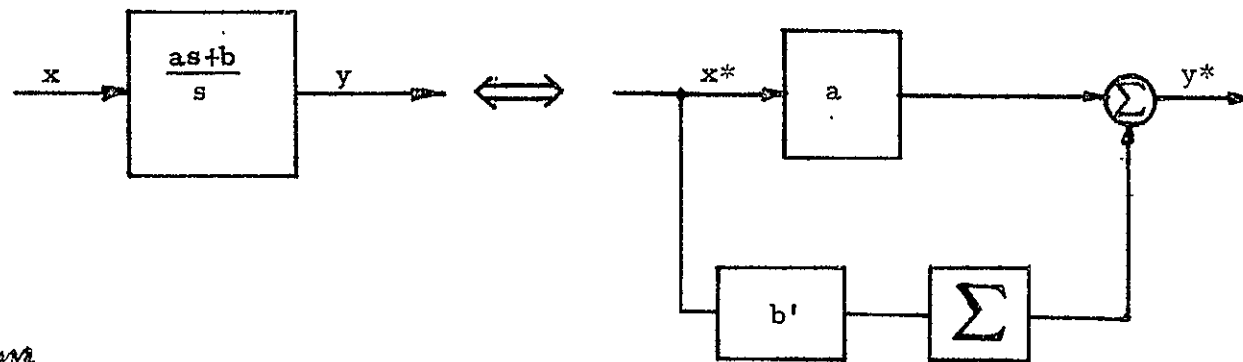
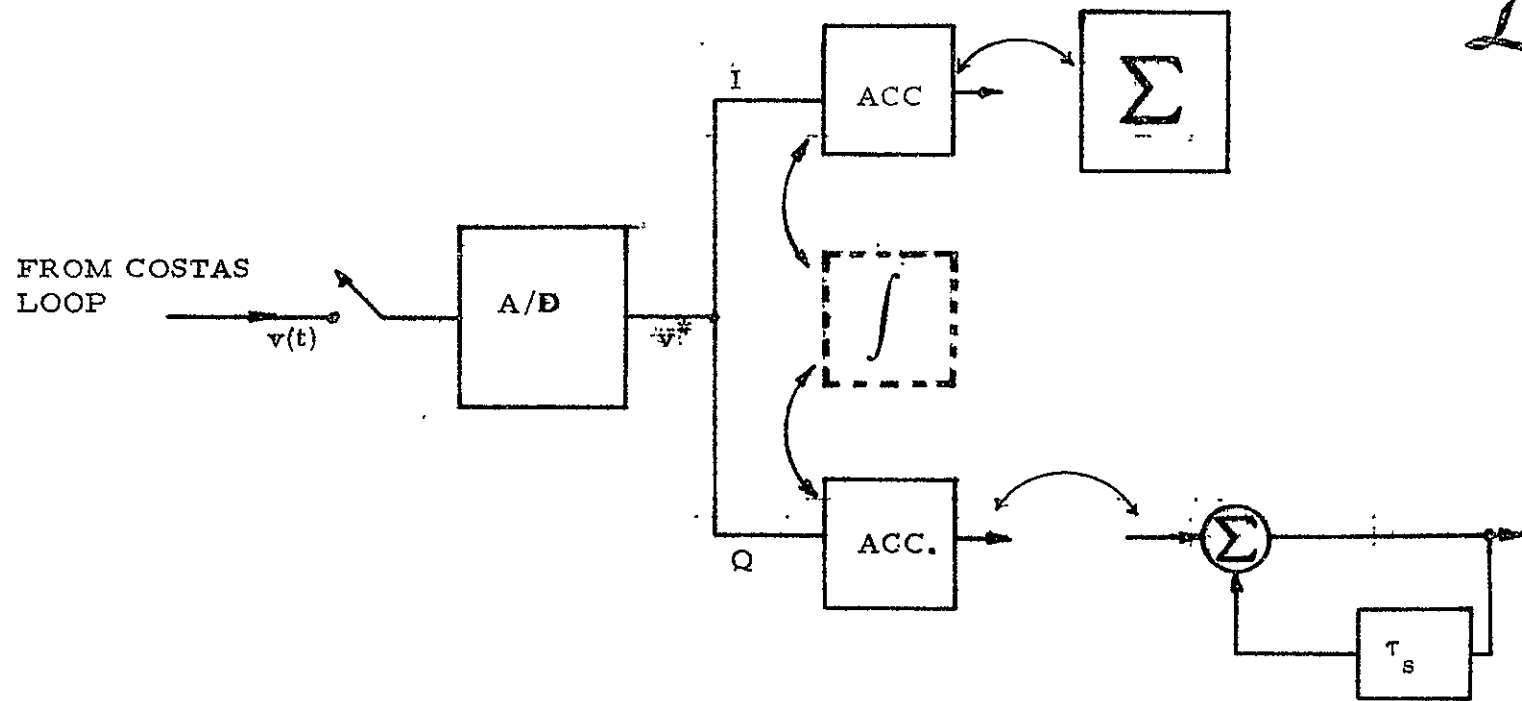
The absence/presence and direction of transitions over half symbols are obtained from the inphase arm integrator which, as previously stated, integrates over half symbols. The polarity detection, transition detection, multiplier and delay hardware shown in Figure 4.9 utilize this information to steer the quadrature arm signal in accordance with the just described algorithm. The performance associated with this implementation is analyzed in section 4.3.

The symbol synchronizer hardware implementation of Figure 4.9 appears to be analog in nature. However a digital design is employed for both the hardware and simulation. As shown in Figure 4.12 the input from the Costas loop is sampled, A/D converted and the integrate and dump circuits replaced by accumulators (summers). A numerical controlled oscillator (NCO) is used in place of a VCO. A digital equivalent of an integral plus proportional loop filter is used.

4.3 Synchronizer Math Model and Loop Analysis

In this section the math model for the symbol synchronizer is presented along with an analysis of steady state performance. We begin by noting that the input timing offset, e , is essentially constant over a large number of symbol intervals and that the loop response is very slow with respect to a symbol interval, i.e., the loop bandwidth W_L symbol duration product T is small ($W_L T \ll 1$). In this

LinCom



LinCom

FIGURE 4.12. DIGITAL IMPLEMENTATION.

region of the parameter $W_L T$ the symbol synchronizer of Fig. 4-13 can be modeled as a continuous phase locked loop illustrated in Fig. 4-14. Determining the steady-state performance of this loop relies on finding: (1) the loop cross-correlation function, $g(\lambda)$, and (2) the two-sided spectral density, $S(\omega, \lambda)$, of the equivalent additive noise $n_\lambda(t)$ at the output of the loop nonlinearity, $g(\lambda)$. Based on the assumptions made above, the mean and variance of the error random variable, e_k , (Fig. 4-16) can be determined assuming λ to be fixed. In effect, then, what we do is to consider many records of the discrete random variable e_k at fixed λ and call the average value of this ensemble $g(\lambda)$, and its spectrum $S(\omega, \lambda)$. The actual spectrum $S(\omega)$ of the additive noise is then obtained by averaging $S(\omega, \lambda)$ over the p.d.f. of λ , $p(\lambda)$, which is to be found; i.e.,

$$S(\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} S(\omega, \lambda) p(\lambda) d\lambda \quad (1)$$

The stochastic differential equation of loop operation is therefore given by (see Fig. 4-6 for notation)

$$\dot{\lambda} = \Delta f T - K_v F(p) [g(\lambda) + n_\lambda(t)] \quad (2)$$

and it is from this equation that loop performance is developed.

With reference to Fig. 4-6, characterization of the phase detector characteristic $g(\lambda)$ and the power spectral density $S(\omega, \lambda)$, can be expressed mathematically as follows

$$\begin{aligned} g(\lambda) &\triangleq E_{n, s} \{e_k | \lambda\} \\ S(\omega, \lambda) &= \mathcal{F}[E_{n, s} \{e_k e_{k+m} | \lambda\}] - g^2(\lambda) \\ &= \mathcal{F}\{R(m, \lambda)\} \end{aligned} \quad (3)$$

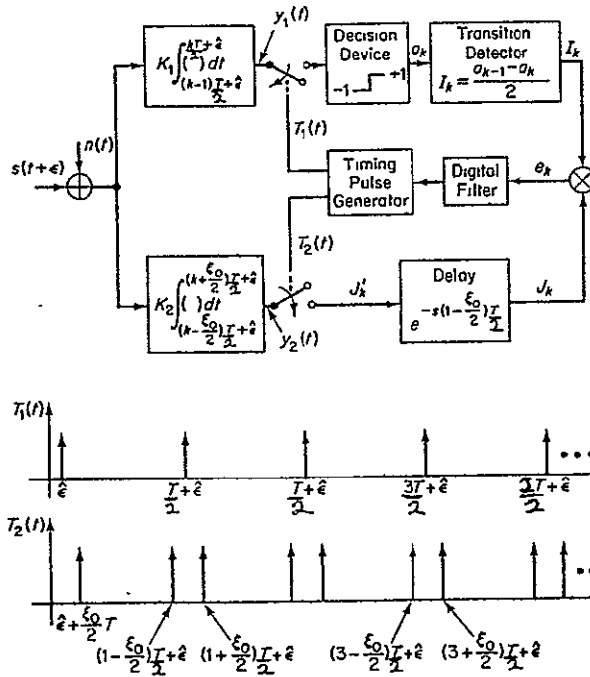
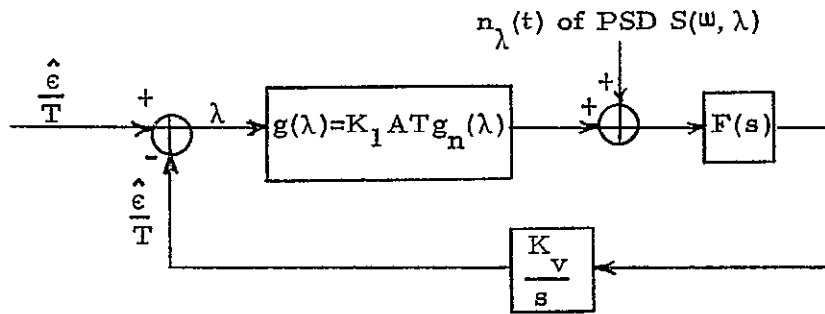


Figure 4-13. The Digital Data Transition Loop (DTTL) for Manchester Encoded Data.



A - Rectangular Pulse Height

T - Symbol Duration

$n_{\lambda}(t)$ - Equivalent Timing Noise

$g_n(\lambda)$ - Normalized Phase Detector Characteristic

K_v - VCO Gain

Δf - Frequency Offset Between Incoming Clock and Local Clock

Figure 4-14. Equivalent Mathematical Loop Model.

where $E_{n,s}$ represents the conditional expectation on λ both with respect to the noise and signal (symbol sequence) and the symbol $\mathcal{F}$ denotes the discrete Fourier transform operation. The autocorrelation function $R(m, \lambda)$ has the following properties:

- (1) $R(m, \lambda)$ has nonzero value only at $m = 0, \pm 1$. Thus the spectrum $S(\omega, \lambda)$ consists of the sum of a constant and a sinusoidal component with period $\omega_0 = 2\pi/T$. Since, as before, it is assumed that $W_L T \ll 1$, it is sufficient then to consider only the value of zero frequency, i.e., $S(0, \lambda)$, and assume a flat spectrum of this value for all ω of interest.
- (2) Since $R(-m, \lambda) = R(m, \lambda)$ for all m ,

$$\frac{S(0, \lambda)}{(K_2)^2 \xi_0 \frac{N_0 T}{4}} = \frac{R(0, \lambda) + 2R(1, \lambda)}{(K_2)^2 \xi_0 \frac{N_0 T}{4}} \triangleq h(\lambda) \quad (4)$$

Since $h(\lambda)$ can be shown (Ref. 3, Chapter 9, p.499) not to be a sensitive function of λ , then we shall present the treatment in what follows in terms of $h(0)$. In its present form (4-2) is a nonlinear stochastic differential equation which can be solved, in the presence of noise, only in the statistical sense. In what follows we shall present the results of the analysis which lead to the characterization of the performance measures given earlier Section 4-1.

For design purposes the linear model will be valuable for predicting performance parameters in the tracking mode of loop operation. If K_g is defined as the slope of $g_n(\lambda)$ (see Fig. 4-16) at $\lambda = 0$ and one assumes that $\lambda \ll 1$ then

$$g_n(\lambda) \approx K_g \lambda \quad (5)$$

and the linearized baseband loop model is shown in Figure 4-15 .
From this figure one readily sees that the closed loop transfer
function is given by

$$H_{\lambda}(\omega) \triangleq \frac{AKF(s)}{s + AKF(s)} \quad (6)$$

where the open loop gain is defined via

$$AK \triangleq K_1 K_g A T K_v \quad (7)$$

Contrast this with that of a sinusoidal PLL, Ref. 2, Chapters 3 and 4, for which the open loop gain is given by $AK = K_1 K_m K_v$. Thus K_v for the PLL is analogous with $K_v T$ for the symbol sync loop, the phase detector gain K_m of the sinusoidal PLL is analogous to K_g in (7) for the symbol sync loop while K_1 in (7) is analogous to the rms value of the VCO reference of a PLL. The linear theory which allows us to define loop parameters will be given at a more convenient point later.

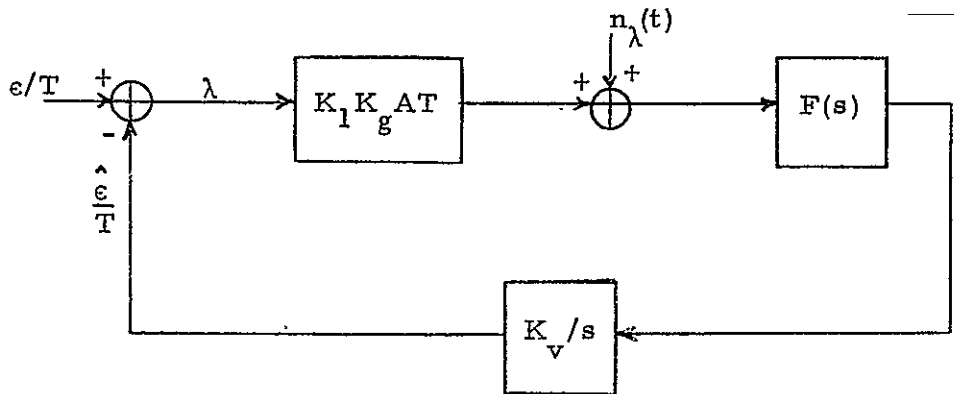


Figure 4-15. Linearized Equivalent Loop Model for Arbitrary Loop Filter.

4.3.1 Phase Detector Characteristics for Sync Acquisition and Tracking

Owing to the fact that the half-symbols are correlated in Manchester data, operation of the phase detector is not the same as that for NRZ data; however, the normalized phase detector characteristic

$$g_n(\lambda) \triangleq \frac{g(\lambda)}{ATK_1} \quad (8)$$

is a function of several basic parameters, viz.,

Energy per symbol-to-noise ratio

$$R_D = R_d \triangleq ST/N_0 = A^2 T/N_0$$

Normalized window width, $\xi \leq \frac{1}{2}$

Data transition probability, $2pq$

Normalized Symbol Sync error

$$\lambda = (\epsilon - \hat{\epsilon})/T$$

Prefilter Characteristic

The normalized phase detector characteristic is illustrated in Figs. 4-16 and 4-17 as a function of the transition density and window width for R_d equal infinity. Unlike the DTTL phase detector for NRZ, notice that the development of $g_n(\lambda)$ has two components. Component $g_1(\lambda)$ is due to the midsymbol transitions which occur with probability 1 while component 2, $g_2(\lambda)$, is due to the between symbol transitions which occur with probability $2pq$. With $2pq$ the probability of a data transition in an NRZ waveform, then the probability of a transition P_t in the Manchester code is

$$P_t \triangleq \frac{1}{2}(1) + \frac{1}{2}(1-2pq) = 1-pq \quad (9)$$

This fact is evident in Fig. 4-16. Clearly, the slope of $g_n(\lambda)$ is a more sensitive to window width variations than it is to transition density. Another property worth noting about the phase characteristic is that it changes sign at $\lambda = \frac{1}{4}$ so that the correlation voltage developed about this point cannot be used as an indicator or supervisor of whether the loop is locked or not. In addition, the phase detector characteristic has normalized stable points at $0, \pm 1/2, \pm 3/2, \dots$ so that the loop can lock one-half of a symbol away from the true zero crossing of the clock by multiples of one-half symbols. This necessitates the need for an ambiguity resolution circuit which can supervise the synchronizer as to the correct clock phase to disseminate to the data demodulator (decoder).

As seen from Figs. 4-16 and 4-17, the phase detector characteristic (S-curve) developed in the Shuttle bit synchronizer is a sensitive function of signal-to-noise ratio ST/N_0 , data transition density, $2pq$, normalized bit sync error $\lambda \triangleq [\epsilon - \hat{\epsilon}]/T$, the spectral roll-off and bandwidth of the pre-detection filter and the window width ξ_0 . The effects of offsets due to phase detector imbalance, viz., unbalance with respect to the reference timing and waveform generating circuitry, and offsets dependent on SNR, are minimized due to the digital mechanization of the phase detector.

4.3.2 Variations in the Slope of the Phase Detector Characteristic for Acquisition and Tracking

In predicting loop performance over the dynamic range of the signal-to-noise ratio and transition densities expected, the normalized slope

$$K_g \triangleq \left. \frac{\partial g_n(\lambda)}{\partial \lambda} \right|_{\lambda=0} = \frac{1}{K_1 AT} \frac{\partial g(\lambda)}{\partial \lambda} \quad (10)$$

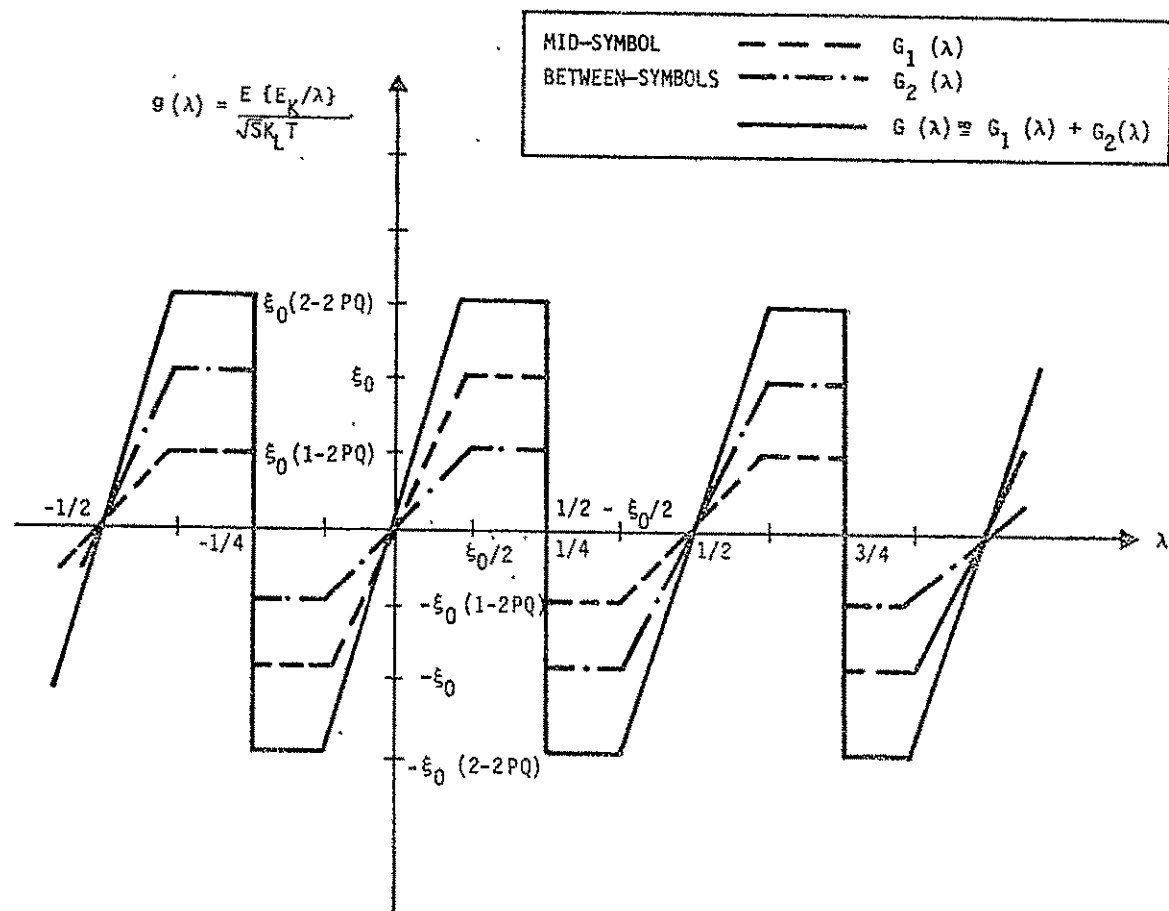


Figure 4-16. Twice Bit Rate NRZ Phase Detector Characteristics (Noise Free).

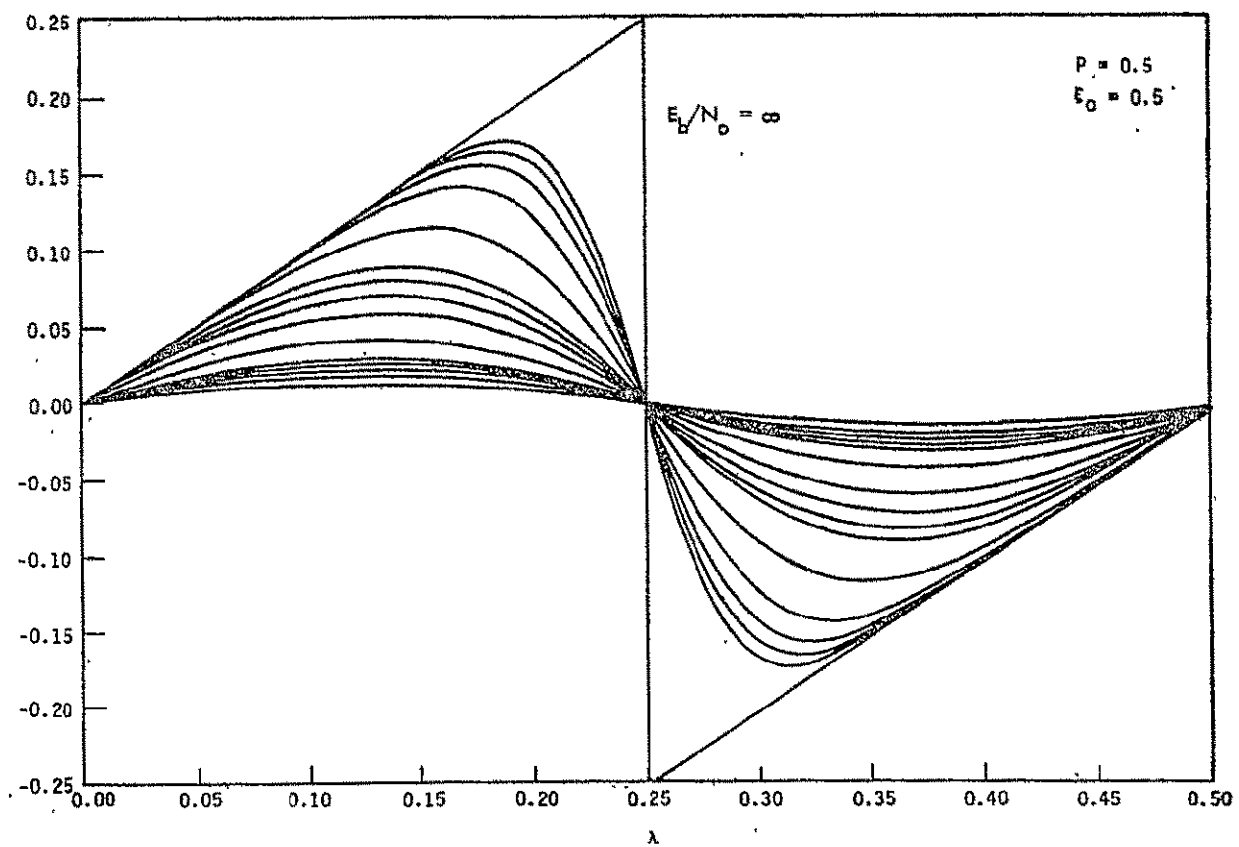


Figure 4-17. Twice Bit Rate NRZ Phase Detector Characteristics.

of the S-curve $g_n(\lambda)$ is of interest. This has been shown to be given by

$$K_g \triangleq (1-pq)\text{erf}(\sqrt{R_d}/Z) - 2pq\xi_0\sqrt{R_d/2\pi}\exp(-R_d/2) \quad (11)$$

Tables 4-3 and 4-4 demonstrate the variations in K_g for end point signal to noise ratios of $R_d = -5$ dB and 7 dB and two design point values of $2pq$. Table 4-3 is for sync acquisition with $\xi_0 = 0.50$ while Table 4-4 is for tracking with $\xi_0 = 0.50$.

R_d ; dB	$2pq$	K_g	AK_g
-5	0.1	0.3857	0.3857
-5	0.9	0.1441	0.1441
7	0.1	0.9215	3.68
7	0.9	0.5005	2.00

Table 4-3. Variations in Normalized Slope K_g for $\xi_0 = 0.5$.

R_d ; dB	$2pq$	K_g	AK_g
-5	0.1	0.376	0.376
-5	0.9	0.0595	0.0595
7	0.1	0.918	1.504
7	0.9	0.47	1.88

Table 4-4. Variations in Normalized Slope K_g for $\xi_0 = 0.25$.

4.3.3 Design Equations for Linearized Second-Order Loop

Design of the symbol synchronizer readily proceeds on the basis of the linear phase-locked loop theory. This requires the establishment of loop filter time constants, gains, etc. in terms of the loop bandwidth, W_L , or natural frequency, ω_n , loop damping ζ and open loop gain AK . In particular if the loop filter is of the form

$$\begin{aligned} F(s) &= \frac{1 + \tau_2 s}{1 + \tau_1 s} \simeq \frac{1 + \tau_2 s}{\tau_1 s} \\ &= \frac{\tau_2}{\tau_1} + \frac{1}{\tau_1 s} = F_0 + \frac{1}{\tau_1 s} \end{aligned} \quad (12)$$

for $\tau_2 \ll \tau_1$, then the loop bandwidth is given by (Ref. 2, Chapter 4)

$$W_L = 2B_L = \frac{4\zeta^2 + 1}{2\tau_2} = \omega_n \left(\zeta + \frac{1}{4\zeta} \right) \quad (13)$$

where ζ is the loop damping and ω_n is the loop natural radian frequency. Here

$$\zeta = \sqrt{\frac{AK\tau_2^2}{4\tau_1}} \quad (14)$$

where

$A = \sqrt{S}$ = Bit Stream Voltage Level

TK_v - VCO gain

K_g - Phase Detector Gain

$AK K_1 K_v TF(s)/s$ - Open Loop Transfer Function

$AK = AK_g K_1 K_v$ - Open Loop Gain

The closed loop transfer function $H(s)$ to be synthesized is found from (8) to be

$$\begin{aligned}
H_{\lambda}(s) &= \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\
&= \frac{1 + \left(\frac{4\zeta^2 + 1}{W_L}\right)s}{1 + \left(\frac{4\zeta^2 + 1}{W_L}\right)s + \frac{1}{2\zeta} \left(\frac{4\zeta^2 + 1}{W_L}\right)^2 s^2} \quad (15)
\end{aligned}$$

where K_g is defined in (11).

In the proposed mechanization the loop filter consists of a direct path from the phase detector output to the VCO with a gain of $F_0 = K_d$ and an integrator path through an up-down counter with a gain of K_2/T_s v/rad. Since synthesized is of the form

$$F(s) = K_d + \frac{K_2}{Ts} = K_d \left(1 + \frac{K_2}{K_d Ts} \right) \quad (16)$$

$$\begin{aligned}
F_0 &= K_d \triangleq \frac{\tau_2}{\tau_1} ; \quad \frac{1}{\tau_1} = \frac{K_2}{T} \\
\tau_2 &= K_d T / K_s \quad (17)
\end{aligned}$$

so that

$$\zeta = \frac{1}{2} \sqrt{\frac{(AK_1 K_g K_v T) K_d^2 T}{K_2}} \quad (18)$$

Furthermore, we have the convenient design formulas

$$\begin{aligned}
K_d &= \frac{8\zeta^2 W_L}{AK_1 K_g K_v T (4\zeta^2 + 1)} = \frac{2\zeta\omega_n}{AK_1 K_g K_v T} = \frac{2\zeta\omega_n}{AK} \\
\frac{K_2}{T} &= \frac{\omega_n^2}{AK_1 K_g K_v T} = \frac{\omega_n^2}{AK} \quad (19)
\end{aligned}$$

for use in designing the digital loop filter. Furthermore, with

$$K_d = 2^{-L}; K_2 = \frac{1}{2^{I-J}} \quad (20)$$

and L, I, J integers, then the closed loop transfer function of (8) becomes

$$H_\lambda(s) = \frac{AK_1 K_g TK_v 2^{-L} s + AK_g K_v 2^{J-I}}{s^2 + AK_1 K_g TK_v 2^{-L} s + AK_1 K_g K_v 2^{J-I}} \quad (21)$$

From this transfer function it is obvious that

$$\begin{aligned} 2\zeta\omega_n &= AK_1 K_g TK_v 2^{-L} \\ \omega_n^2 &= AK_g K_v 2^{J-I} \\ \zeta &= (2^{I-J-L-1})\omega_n T \\ B_L T &= \frac{2^{2I-2J-2L} (\omega_n T)^2 + 1}{2^{I-J-L+2}} \\ 2^{-L} &= \frac{2\zeta\omega_n}{AK_1 K_g K_v T} \\ 2^{-I} &= \frac{\omega_n T}{\zeta 2^{J+L+1}} = \frac{(\omega_n^2)}{AK_1 K_g K_v 2^J} = \frac{(\omega_n^2)}{AK_1 K_g K_v} \end{aligned} \quad (22)$$

which are useful in design of the digital loop filter. Since K_g varies with K_d , the window width and the transition density, we see that this causes a corresponding variation in loop bandwidth and damping.

Because the phase-detector gain, say K_g , varies with the signal-to-noise ratio (SNR) $R_d = ST/N_0$, transition density and window width, the loop's bandwidth B_L , damping factor ζ and loop

SNR also vary. Using (13) the single-sided loop bandwidth at R_d , relative to the design point of $R_d = -5$ dB, varies in accordance with, Ref.

$$\frac{B_L(R_d, \xi_0, pq)}{B_L(-5\text{dB}, \xi_0, pq)} = \frac{4\zeta^2(R_d, \xi_0, pq) + 1}{4\zeta^2(-5\text{dB}, \xi_0, pq) + 1} \quad (23)$$

4.3.4 Symbol Sync Jitter Performance

The normalized symbol sync jitter σ_λ as a function of C/N_0 is illustrated in Fig. 4-18 for various values of the loop bandwidth. Since B_L varies with signal-to-noise ratio we have superposed a set of design curves (dashed) which show how performance degrades as the bandwidth opens from the design point values of $B_L = 200, 400$ and 1000 Hz. The solid curves indicate symbol sync jitter performance as a function of C/N_0 with the loop bandwidth held fixed. As seen from this figure, system jitter performance is not extremely sensitive to the opening of the loop bandwidth as the signal strength rises.

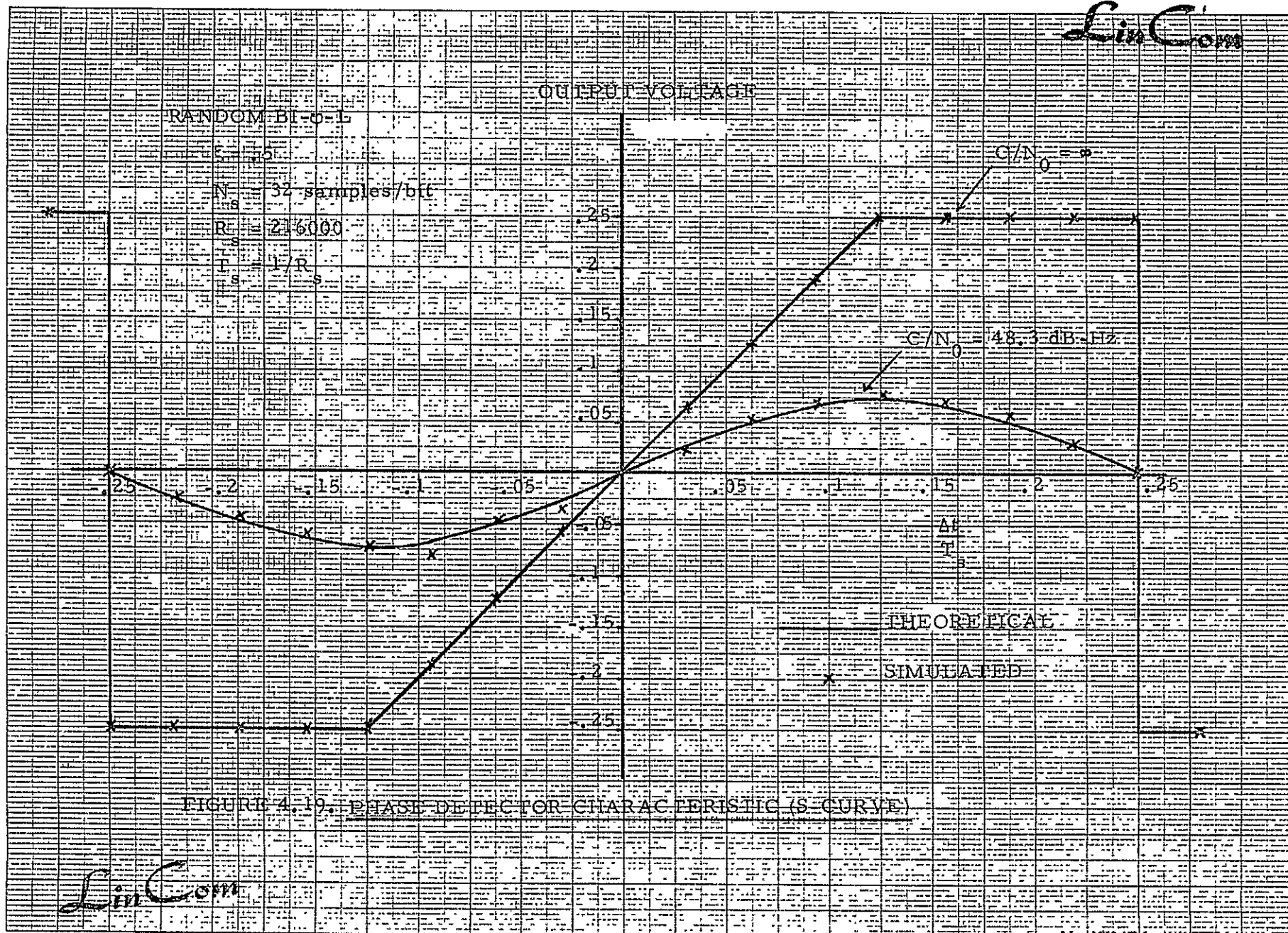
4.4 Simulation Verification

A series of experiments were performed to verify simulated results against those analytically predicted.

4.4.1 Phase Detector Characteristic

A phase detector characteristic is generated by breaking the loop and slightly offsetting the VCO from the incoming timing. The theoretical "S" curve characteristic that would be measured at the output of a lowpass filter is shown as the solid curve in Figure 4.19. With no noise, simulated and theoretical results are in excellent agreement. Rather good agreement exists in deep noise with the slight discrepancies due to measurement inaccuracies. Note the significant reduction of the S curve slope in deep noise. This reduction is associated with the

LinCom

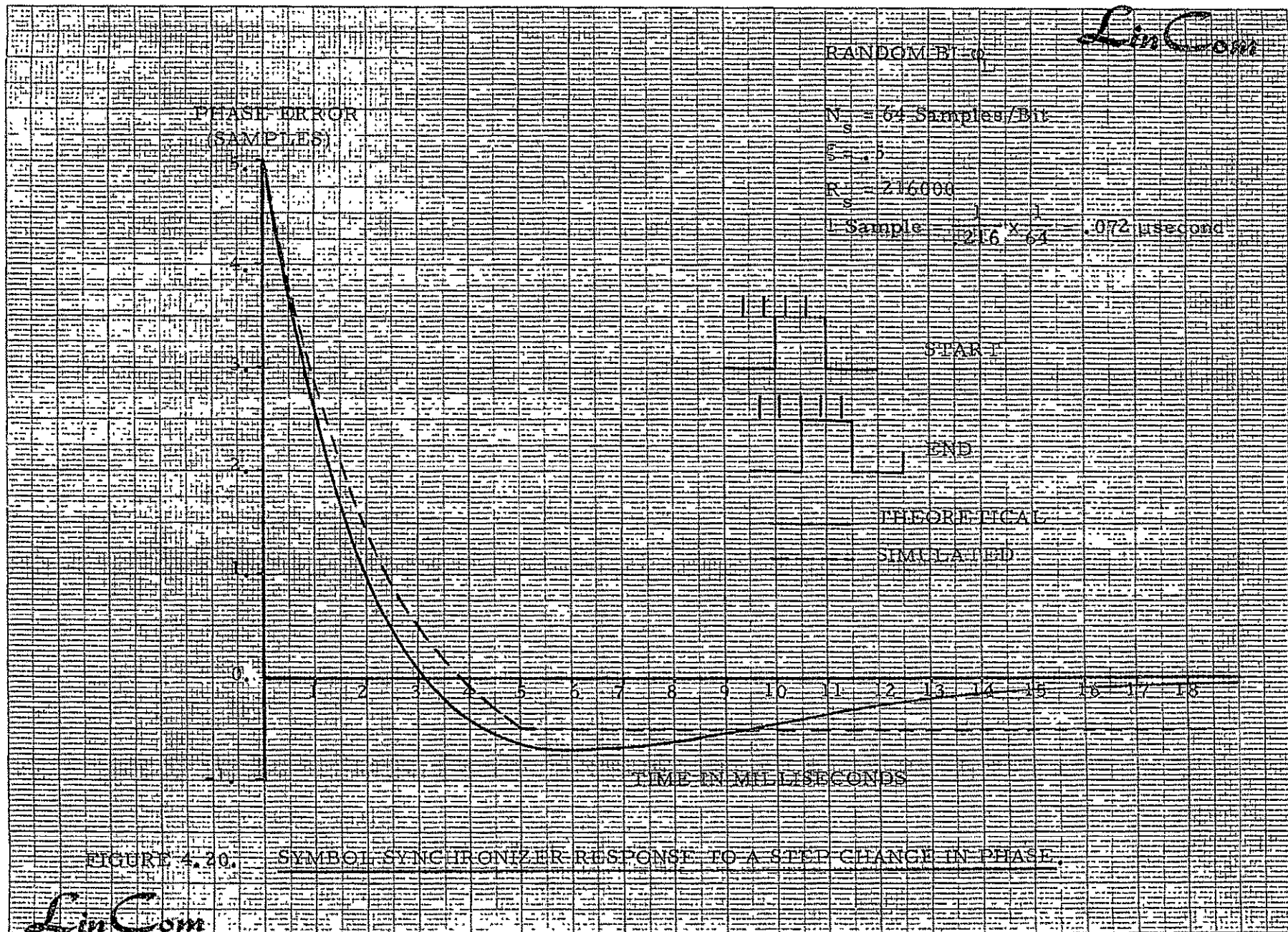


large numbers of errors that occur in determining the transition directions used to steer the quadrature arm signals.

It is also to be noted that the S curve characteristic exhibits an ambiguity each half symbol. Logic associated with the data detection process selects our true end of symbol time from the mid-symbol time by counting mid and end of symbol transitions. With random data there will be twice as many mid symbol transitions as there are start of symbol transitions. Without this feature the data detected would have no meaning as its value would represent the last half of the previous symbol and the first half of the present one.

4.4.2 Transient Response

Theoretical transient response of the phase error to a step change in phase and a step change in frequency are shown as solid curves in Figures 4.20 and 4.21. A linear model was used in the calculation. The simulated results are shown as dotted curves. A phase offset of $\frac{5}{64}$ of a symbol and a doppler offset of 50 symbols per second kept the transient responses in the linear range of the S curve but even so, note the relatively large discrepancies between theoretical and measured performance. Quantizing effects associated with the sampling process caused this discrepancy. Also note the half sample error in steady state. This half sample error can be explained using the waveforms sketched to the right side of the transient response waveforms of Figure 4.20. At $t = 0$ the symbols were sampled as shown by the start figure (midway between possible transitions). The end figure represents steady state with one of the samples aligned with the symbol transition. This resolution problem does not significantly degrade performance, however, as noise or symbol filtering tends to better center the samples around the transition points.



LinCom

PHASE ERROR
(SAMPLES)



TIME IN MILLISECONDS

RANDOM B: $\rho=1$

$N_B = 64$ Samples/Bit

$\rho = 0.5$

$R_B = 216000$

1 Sample $= \frac{1}{216} \times \frac{1}{64} = .072$ /second

$\Delta f = 50$ Symbols/Second

THEORETICAL

SIMULATED

FIGURE 4-21. SYMBOL SYNCHRONIZER RESPONSE TO A STEP CHANGE IN FREQUENCY.

LinCom

4.4.3 Performance In Noise-Jitter

A theoretical plot of RMS jitter (measured as a percent of the symbol) versus loop bandwidth is shown in Figure 4.22 , for a 48.3 dB-Hz C/N_0 condition. Note the good agreement with the simulated datum points. Jitter over the wide range of bandwidths shown is quite low, being only 1.5% at our system design point loop bandwidth of 30 Hz. As can be seen from Figure 4.22 only at large bandwidth values will the jitter become appreciable. Note the excellent agreement between simulated and theoretical data.

4.4.4 Performance In Noise - Error Rate

A curve of theoretical symbol error rate vs C/N_0 is shown in Figure 4.23 (ideal loop) . Measured error rates, for various loop bandwidths, are plotted as datum points for an input C/N_0 of 48.3 dB-Hz. At our design point loop bandwidth of 30 Hz there will be very little measurable degradation in performance associated with the SSS. In fact, increasing the loop bandwidth to 200 Hz would only degrade performance slightly ($\sim .45$ dB).

4.5 Software

The Symbol Synchronizer Subsystem is simulated in subroutine SSS. It is designed to replace subroutine BSS of the Phase I effort. The input/output interface parameters for SSS are shown in Figure 4.24. A more detailed description of SSS can be found in Appendix B. There are seven parameters capable of being varied via a single data card. They are: loop bandwidth, loop damping factor, samples/bit, symbol rate, doppler, window size, and loop option. There are two inputs originating in other programs. The first an output from CSS, is the waveform at the lower arm mixer of the Costas loop. The second is a mode control set from the main calling routine. Mode = 1 or 2 initialize SSS while MODE = 3 controls the normal loop tracking and data detection functions. Program

RMS JITTER (% of a Symbol)

LinCom

46 1513

10 X 10 TO THE CENTIMETER 18 X 25 CM
KEUFFEL & ESSER CO. MADE IN U.S.A.

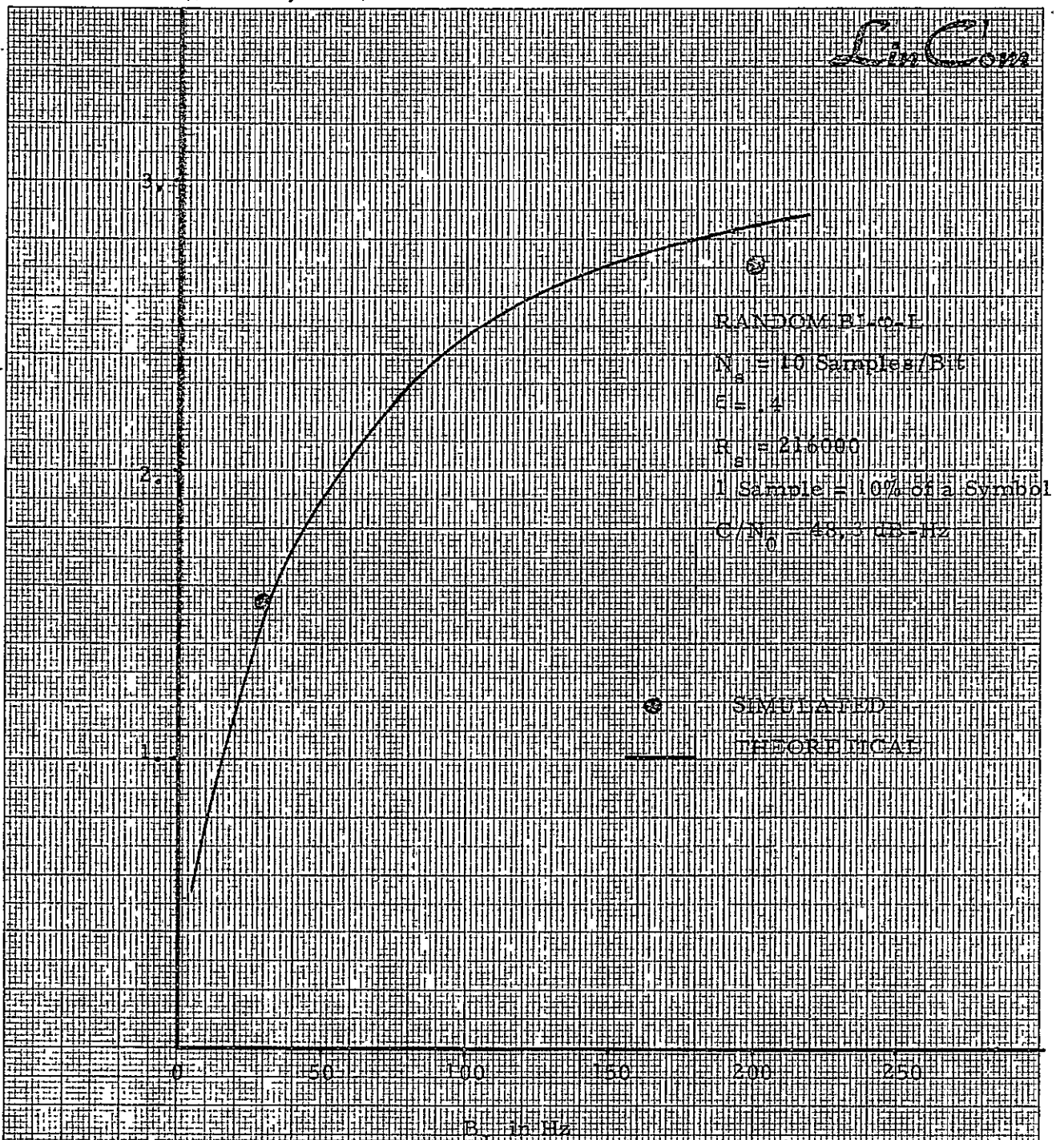


FIGURE 4.22. RMS JITTER AS A FUNCTION OF LOOP BANDWIDTH.

LinCom

LinCom

SYMBOL
ERROR RATE

THEORETICAL

RANDOM BPSK

$N_s = 10$ Samples/Symbol

$M = 4$

$R_s = 216000$

$B_L = 200$ Hz

$B_L = 30$ Hz

$B_L = 400$ Hz

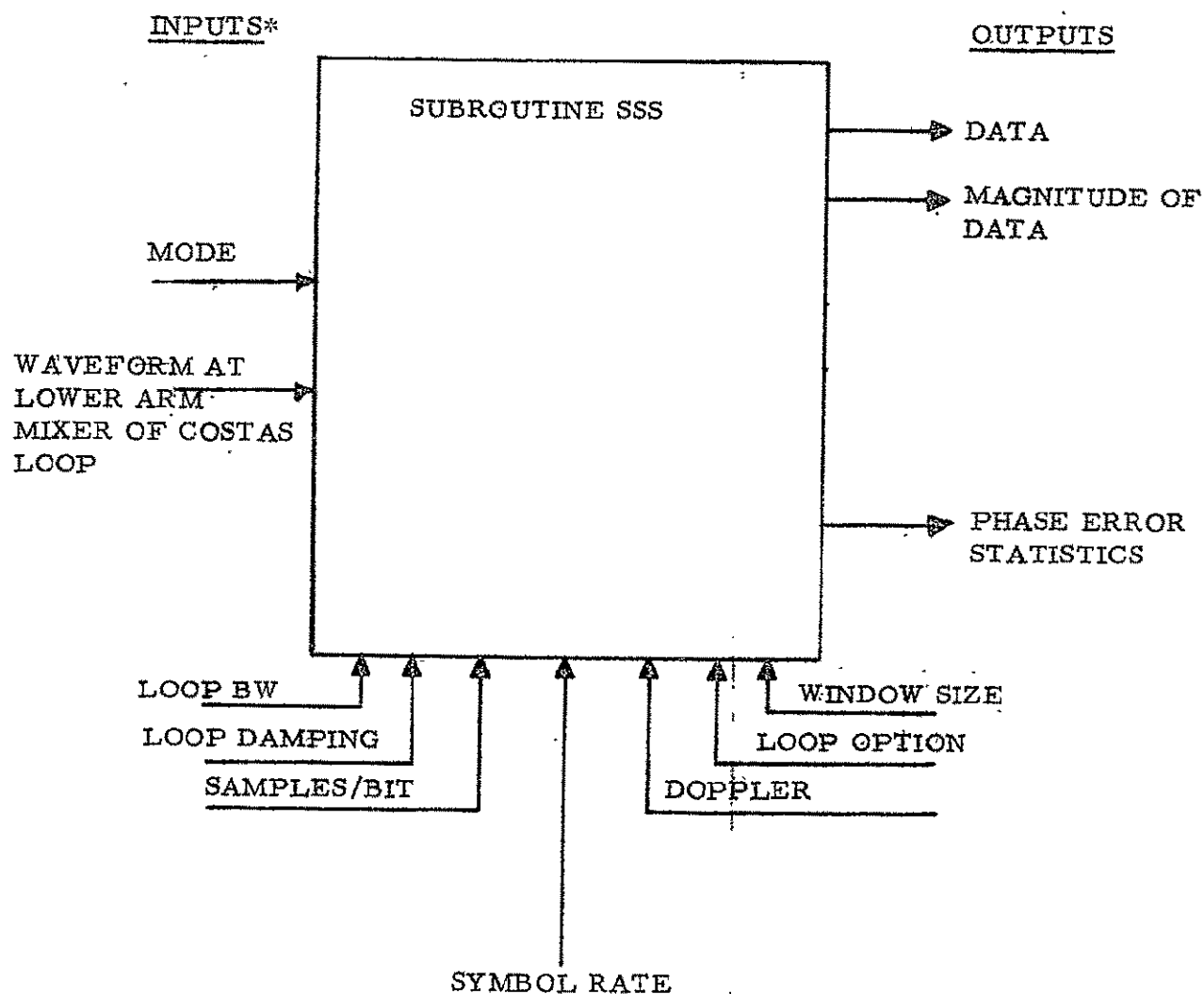
$C/N_0 = 48.3$ dB-Hz

(.45 dB)

C/N_0 dB-Hz

FIGURE 4.23. SYMBOL ERROR RATE VS C/N_0

LinCom



EXTERNAL PARAMETERS[†]

*Inputs from Other Programs

†Read from Data Cards

FIGURE 4.24. INPUT/OUTPUT PARAMETERS FOR SUBROUTINE SSS.

outputs are the sign and magnitude of detected symbols which are used by program BER in measuring error rate. Jitter statistics are periodically printed out as the run progresses.

References

1. Lindsey, W. C., Tausworthe, R. C., "Digital Data-Transition Tracking Loop," Jet Propulsion Laboratory, Pasadena, Calif., SPS 37-50, Vol. III, April, 1968, 272-276.
2. Lindsey, W. C., Synchronization Systems in Communication and Control, Prentice-Hall, 1972, Englewood Cliffs, N.J.
3. Lindsey, W. C., Simon, M. K., Telecommunication Systems Engineering, Prentice-Hall, 1973, Englewood Cliffs, N.J.
4. Batson, B., Cellier, A., Lindsey, W. C., and Vang, H., "An All-Digital Manchester Symbol Synchronizer for Space Shuttle," Proc. of the National Telecommunications Conf., 1974, San Diego, Ca.

SECTION 5

SYSTEM INTEGRATION

5.0 System Test Facility

A discussion of the integration of the Modulator and Channel Subsystem, the RF front end and PN Subsystem, the Carrier Recovery Subsystem, and the Symbol Synchronizer Subsystem into the composite spread spectrum receiver and test facility will be presented in this section. A detailed description of all but the Symbol Synchronizer Subsystem can be found in Sections 2 thru 4 of the Volume II, Phase I report. This section parallels section 5 of the Volume II report.

The above mentioned test facility includes the modulator, signal sources, RF wideband white noise generator, oscilloscopes, jitter and error rate measurement equipment, and a teleprinter. The printer is used to output the results of test measurements and system status. The above software facility is identical to that found in an actual hardware test bed with one exception, parameter settings are via data cards and not by panel switches.

5.1 Software Integration

A tabulation of all routines defined, developed, documented, and used during the Phase I study are tabulated in Table 5.1A. Modification and changes made during the Phase II development effort are listed in Table 5.1B. Subroutine SSS, defined in section 4, replaces subroutine BSS. Changes to subroutines RFSS, PNMON and DLL will be discussed in section 5.2. A new subroutine, DATGEN, has been added and will also be described in section 5.2. The main control program, MAIN, that drives all the subroutines is described in detail in Appendix C-1. This program is suitable for evaluating the tracking and acquisition behavior of our receiver. The main program is partitioned into two parts, initialization, and acquisition and tracking.

TABLE 5.1A. PHASE I SOFTWARE MODULES

MAIN - Driving Program
PNCOR - RF filter and cross correlation function
RFSS - Modulator and channel
RFAGC - Wideband RF AGC
PNMON - PN microprocessor logic
PNACQ - PN acquisition hardware
DLL - Delay locked loop
IF2 - 2nd IF Filter
CSS - Costas Loop, AGC, and sync circuit
MONTOR - Carrier Microprocessor logic
CTR - Carrier jitter measurement
BER - Symbol error rate measurement
BSS - Bit synchronizer
AANDN - Noise source
RANDN - Gaussian noise generator
URAN - Uniform number generator
TINORM - Randn support software

TABLE 5.2B. PHASE II MODIFICATIONS

SSS - Symbol Synchronizer Subroutine (Replace BSS)
MAIN, RFSS, DLL, PNMON (Changed)
DATGEN - Modulator Data Source (Added)

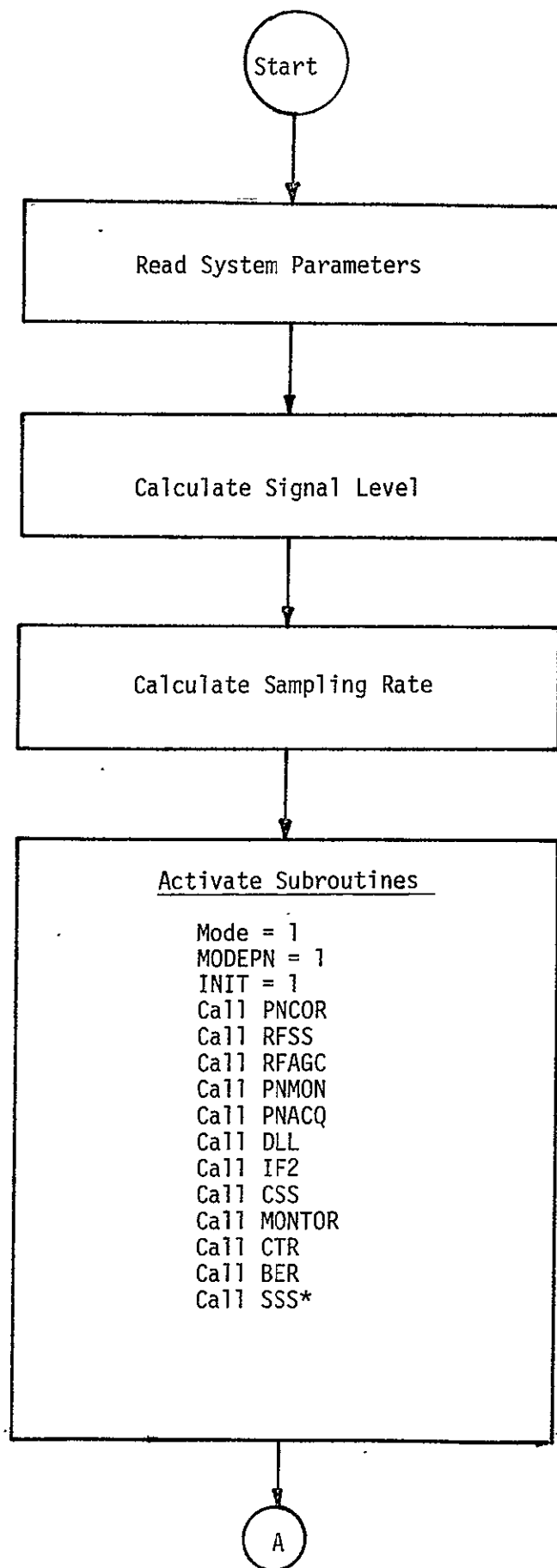
5.1.1 Initialization

The initialization sequence is shown in Figure 5.1. At the start of program execution the system parameters are read from data cards. The modulator output signal level is then calculated using the inputted C/N_0 and an internally fixed value of 3.7×10^{-6} watts/Hz for N_0 . The system sampling rate is also calculated as a multiple of the specified data rate and samples per bit. The following subroutines, in the order shown, will then be called:

PNCOR, RFSS, AANDN, URAN, RANDN, TINORM, DATGEN,
RFAGC, PNMON, PNACQ, DLL, IF2, CSS, MONTOR, CTR,
BER, SSS.

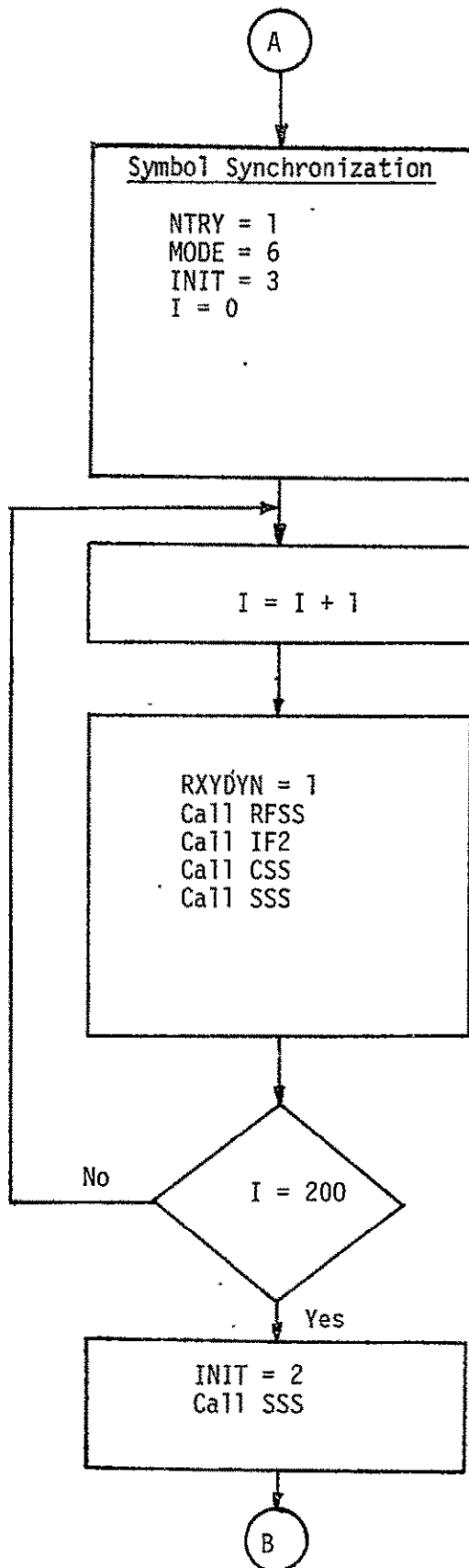
Please note that AANDN, URAN, RANDN, TINORM and DATGEN are not shown in Figure 5.1 but are activated via the call to RFSS. The subroutines associated with the Carrier Recovery and Symbol Synchronizer Subsystems are initialized by setting INIT and MODE equal to unity. The RF front end and PN Subsystem are activated by setting MODEPN to one. Activation includes the evaluation of the Z-form coefficients associated with the various filters, setting initial conditions such as random code phases and carrier frequency, adjusting loop gains, etc. It is suggested that the calling order shown be followed as some subroutines depend on setting parameters with values calculated in other programs.

To reduce computer execution time the symbol synchronizer timing is acquired prior to PN and carrier acquisition. This is accomplished by passing the correlated signal waveform ($RXYDYN = 1$) through the IF filter and Costas loop with a 1000 Hz SSS bandwidth. Noise is not injected ($NTRY=1$) during this process. As shown in Figure 5.2 200 bits of data are transmitted to allow the subsystem



*1000 Hz Loop
Bandwidth

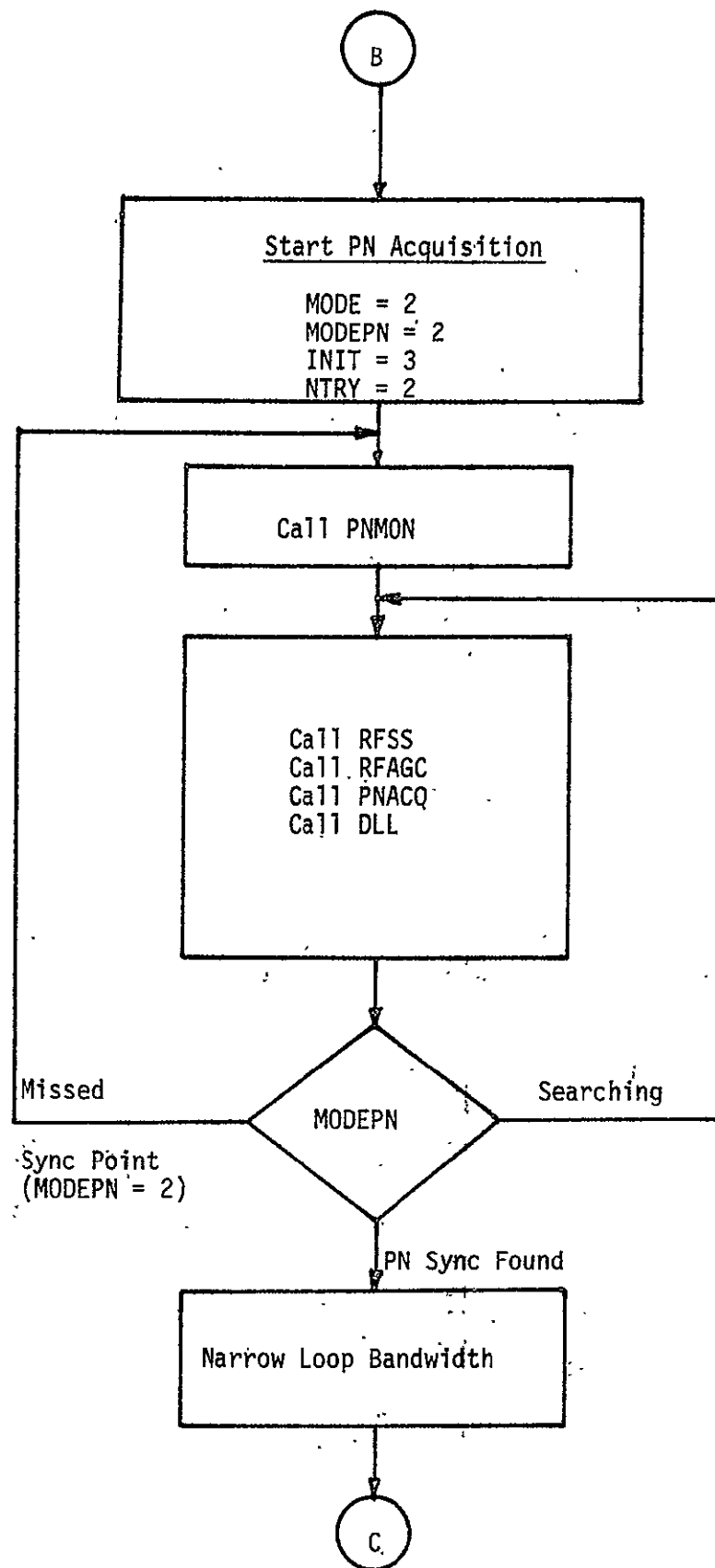
FIGURE 5.1. PROGRAM INITIALIZATION.

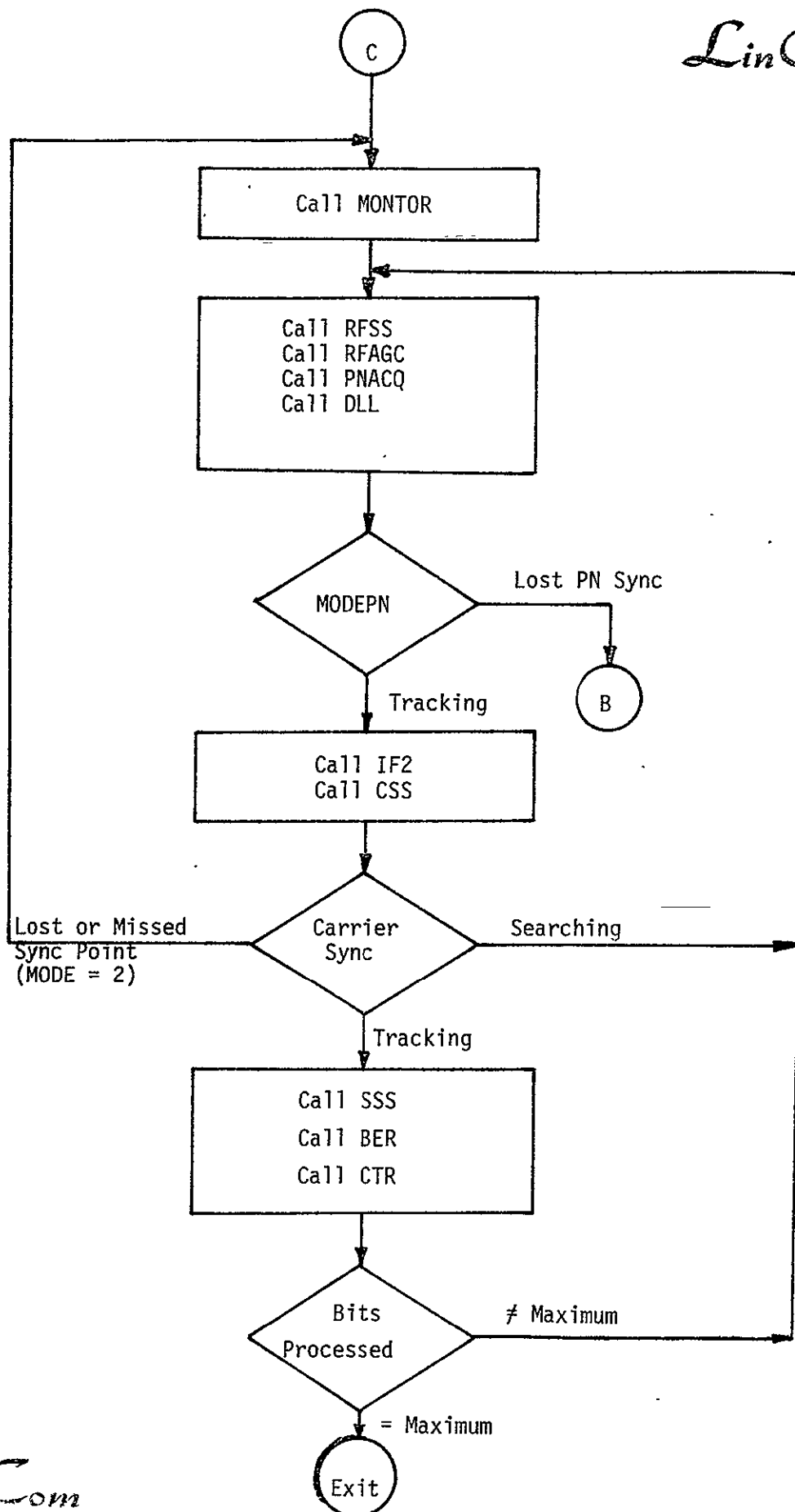
FIGURE 5.2. SYMBOL SYNCHRONIZER ACQUISITION.

transients to settle. Note that each call to RFSS generates a signal waveform over a one bit interval. In the figure INIT = 3 activates the Symbol Synchronizers' timing and MODE=6 disables the sweep control logic associated with CSS. After acquisition INIT is set to 2 and subroutine SSS called once again. This switches the loop bandwidth from a 1000 Hz to that specified via an input parameter.

5.1.2 Acquisition and Tracking

Once symbol synchronization is completed the receiver is ready to acquire the spreading waveform and transmitted carrier and enter the track mode. The program calling sequence which accomplishes this is shown in Figure 5.3. PN acquisition takes place first with the Carrier Recovery Subsystem not operating. The parameters MODE and MODEPN are set equal to 2. Letting NTRY=2 adds the channel noise. Setting INIT=3 corresponds to the normal operating mode of the Carrier and Symbol Synchronizer Subsystems. Subroutine PNMON is then called. When entered in MODEPN=2 this routine steps the local PN code phase until it is nearly aligned with the transmitted PN pattern, gathering sync statistics in the process. At this point the signal plus noise waveform is generated via RFSS and passed through the software (RFAGC, PNACQ, DLL) simulating the PN hardware on a sample by sample basis. At the sync decision times subroutine PNACQ calls PNMON (not shown in the figure) which then determines the absence/presence of sync. The calling sequence continues until sync is found and verified or until the local code phase has moved passed the point where this is possible. When the latter occurs the MODEPN flag is reset to 2 via the PNACQ-PNMON call which automatically causes PNMON to be re-entered from the main program and the searching through another code cycle with the program not operating. When PN





sync is found the delay lock loop bandwidth is narrowed and the Carrier Subsystem starts its acquisition procedure.

Subroutine MONTOR (see page 2 of Figure 5.3) is entered from the main program with the MODE flag still unchanged. MONTOR then sweeps the local VCO until its frequency is sufficiently close to the transmitted carriers' frequency to allow the Costas loop to synchronize. Once again the signal plus noise waveform is generated and is now passed through both the PN Subsystem and the IF filter (IF2) and Costas loop (CSS) associated with the Carrier Recovery Subsystem. At the carrier sync decision times subroutine CSS calls MONTOR (not shown in the figure) which determines the carrier lock status. The calling sequence continues until sync is found and verified or until the local VCO is swept past the point where acquisition is possible. When the latter occurs (MODE=2) subroutine MONTOR is re-entered from the main program and the local VCO goes through nearly a full sweep cycle with the program not operational, gathering statistics in the process. When the carrier is acquired the receiver enters the tracking mode.

In tracking, the symbol synchronizer and test measurement software are added to the calling sequence described in the preceding paragraph. The full program is now operational on a sample by sample basis and will remain so until the maximum specified data bits are transmitted and detected or until sync is lost, in which case acquisition is reinitiated.

5.1.3 Input Parameters

Four groups of parameters are inputted via data cards. The first data block, associated with the PN Subsystem, is tabulated in Table 5.2. The second parameter grouping, associated with the Carrier Subsystem is listed in Table 5.3. Table 5.4 tabulates

TABLE 5.2 PN SUBSYSTEM INPUT DATA			
Sequence Order	Name	Description	Test Program Value
1	IOPTON	Speed Improvement Option	0
2	LFGNOS	Noise Source Option	1
3	CODDOP	Code Doppler, chips/second	300.
4	THP1	Acquisition Threshold	1.056
5	THP2	Verify Threshold	1.028
6	THP3	Tracking Threshold	1.051
7	BITRAT	Symbol Rate, symbols/second	216000.
8	TS1	Acquisition Integration Time, seconds	.0009
9	TS2	Verify and Track Integration Time, seconds	.0036
10	TMAX	Maximum Specified Sync Time, seconds	60.
11	RFBW	$\frac{1}{2}$ Power RF Filter Bandwidth, Hz	22000000.
12	TAU	RF AGC Time Constant, seconds	.02
13	MTIME	Samples/Bit	10
14	BWPNAC	Despreader IF Filter Noise Bandwidth, Hz	550000.
15	DLLBLL	τ -Dither Loop Noise Bandwidth, Hz	200.
16	DLLZET	Loop Damping Factor	.70711
17	DLLBIF	Loop IF Filter Noise Bandwidth, Hz	550000.
18	SIGNOS	Input C/N ₀ , dB-Hz	48.3
19	RFLOSS	Front End Loss, dB	0.0
20	CHPRAT	Chip Rate, chips/second	11000000.
21	RF	RF Filter Ripple Factor, dB	1.

TABLE 5.3. CARRIER SUBSYSTEM INPUT DATA			
Sequence Order	Name	Description	Test Program Value
1	DOPFRE	Doppler Frequency, Hz	51000.
2	DOPRAT	Doppler Rate of Change, Hz/second	0.
3	STAPHA	Carrier Static Phase, radians	0.
4	TIME1	Sync Integration Time, seconds	.002
5	TIME2	AGC Time Constant, seconds	.002
6	THR1	Acquisition Threshold	.45
7	THR2	Verify and Tracking Threshold	.225
8	FMAX	Magnitude of Maximum Doppler Uncertainty, Hz	55000.
9	SWEPR	Sweep Rate, Hz/second	40000.
10	KNMAX	Maximum Specified Sweep Cycles	5
11	MAXBIT	Maximum Bits Transmitted	6000
12	ZETA	Costas Loop Damping Factor	1.
13	BANDW	Loop Noise Bandwidth, Hz	500.
14	NTIME	Samples/Bit	10
15	BITRAT	Symbol Rate, symbols/second	216000.
16	BIF2	IF Filter Noise Bandwidth, Hz	864000.
17	CAFU	Upper Arm Filter Noise Bandwidth, Hz	324000.
18	CAFL	Lower Arm Filter Noise Bandwidth, Hz	324000.

TABLE 5.4 SYMBOL SYNCHRONIZER PARAMETERS			
Sequence Order	Name	Description	Test Program Value
1	IOPT9	Idealize (LOCK) SSS	0
2	BSSBLL	Loop Noise Bandwidth, Hz	30.
3	BSSZET	Loop Damping Factor	1.
4	BSSDOP	Doppler, Symbols/sec	30.
5	SYMRAT	Symbol Rate, Symbols/sec	216000.
6	NZETA	Window Size, samples	1

TABLE 5.5 SYMBOL SOURCE PARAMETERS			
Sequence Order	Name	Description	Test Program Value
1	ICDGO	Random or Pattern Selection	1
2	ICDG1	Burst Length (Bits)	1
3	ICDG2	Frame Length (Bits)	100
4	ICDG3	Number of Pattern Bits Used	1
5	ICDG4(32)	32 Bit Pattern	11111111 11111111 00000000 00000000

the third parameter grouping which is associated with the symbol synchronizer. Data source parameters, the final group entered, are listed in Table 5.5. Each block of data is called by unique free format read statements. The first read enters PN Subsystem parameters into the program. The second read calls the Carrier Subsystem data. The third read enters the SSS data. The fourth read enters the single parameter ICDGO associated with the modulator data source. If a completely random pattern is not specified by ICDGO then a fifth read statement is used to enter the pattern information desired. The data associated with each block must be punched on cards with the exact sequence indicated in the tables. A typical data card set up is shown in Figure 5.4.

A glance at Tables 5.2 and 5.3 indicates some duplication of parameter values. This occurs because the design is such as to allow each of these subsystems to be simulated separately in which case only one group of complete values are required. When there is duplication, the parameters of Table 5.3 take precedent. The only restriction placed on the input values used is that the samples/bit, MTIME and NTIME, be identical.

5.1.4 Output Parameters

The inputted parameter values along with information on the RF front end will be printed out during program initialization. Once acquisition starts and until the program runs to completion all printouts will take one of two forms. System status messages delineated by asterices and measurement data delineated by dotted lines. Typical outputs are shown in Figure 5.5.

5.1.5 Test Data

The data values shown on the cards in Figure 5.4, identically the same as that tabulated in Tables 5.2, 5.3, 5.4, and 5.5 will produce a system response whose output printout is listed in Appendix

ENDING PAGE BLANK NOT FILMED

DATA CARD
#5

SYMBOL SYNCHRONIZER DATA CARD

DATA CARD
#6

DATA CARD
#7

[illegible]

DATA SOURCE PARAMETERS

FIGURE 5.4 (Cont'd)

```

*****
*
*          *****RUN ABORTED*****
*
* MAXIMUM PN ACQUISITION TIME OF      .00 SECONDS EXCEEDED.
* NUMBER OF TIMES FALSELY IN TRACKING MODE=      0.
* NUMBER OF CHANCES TO ACQUIRE=      0.
* NUMBER OF FALSE HITS=      0.
*
*****

```

FIGURE 5.5A. TYPICAL OUTPUT STATUS MESSAGE.

```

DATA DETECTION STATUS
-----
THE CUMULATIVE SYMBOL ERROR PROBABILITY OVER THE PAST      0 BITS WAS .0000000.
OVER THE LAST      0 BITS THE SYMBOL ERROR PROBABILITY WAS .0000000.

```

FIGURE 5.5B. TYPICAL OUTPUT DATA MESSAGE.

C-2. This test data along with its corresponding output can be used to exercise the delivered software, as required.

5.1.6 Options

A number of options which influence the software's behavior are available. These options are selected either through the input data blocks or by minor modifications to the main program code. Changes in the inputted filter bandwidths and time constants will allow operating the program at data rates other than 216,000 bits/second. In addition there are 4 input parameters which influence system performance. As discussed in Volume II, Phase I, the data parameters RF and RFBW control the selection of the RF filter and BIF2 determines whether the 2nd IF filter is to be bypassed. IOPT9 can be used to idealize the SSS.

Two other parameters LFGNOS and IOPTON have an influence on the computer execution time. As discussed in the Phase I report, LFGNOS selects the white Gaussian noise source. With LFGNOS=1 a large number of noise samples are generated and stored at one time and then used for the duration of the computer run. This speeds up program execution as independent noise samples are no longer generated each sample time (LFGNOS=0). IOPTON=0 is the normal operating mode for which the Manchester data is generated and the exact effects of the realizable IF filters are included. With IOPTON=1 the modulation is removed from the carrier and ideal (rectangular frequency response) filters replace the physically realizable ones actually used. This lowers the sampling rate which in turn, increases program speed. IOPTON=1 is helpful in studying the acquisition behavior and subsystem interactions.

Simple coding mods made to the main program can significantly change the simulation structure. As an example, the PN Subsystem interfaces with the Carrier Recovery Subsystem by only three parameters. The nominal RFAGC gain, GAINN, the instantaneous

AGC gain, GAINI, and the degree of correlation in the spread waveform, RXYDYN. Setting GAINN equal to an arbitrary value such as .1, GAINI equal to GAINN and RXYDYN to unity simulates a coherent BPSK Com-system. All calls to the PN subroutines can be removed from the main program. Other mods are available which isolate the subsystems and the components within the subsystem for further more detailed study.

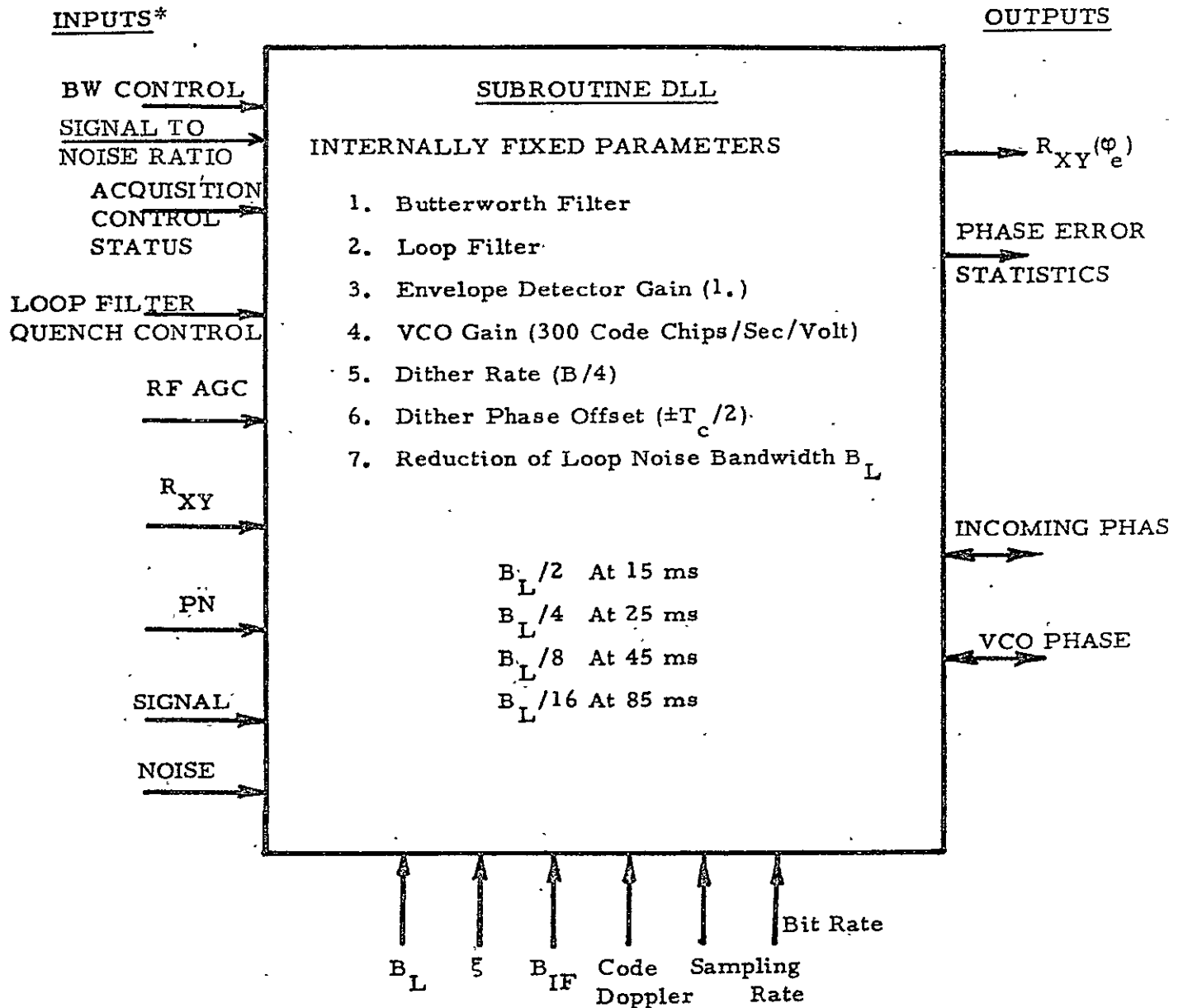
5.2 Subroutines DLL, PNMON, RFSS, and DATGEN

This section serves as documentation for modifications made to the Phase I routines, DLL, PNMON and RFSS and describes the newly developed DATGEN.

5.2.1 Subroutines DLL and PNMON

This software is used to simulate the π -dither loop hardware and the PN search microprocessor. The Phase I program did not model loop bandwidth switching effects on the acquisition to tracking transition. Operationally, after PN sync is found the loop bandwidth is narrowed via a 4 step procedure. It normally requires about 300 milliseconds of real time before loop transients settle. An exact simulation of this settling process requires about 7 minutes of computer execution time and is not really warranted when weighed against the usefulness of the information obtained. The Phase II software carefully switches the bandwidth to its steady state tracking value without significantly perturbing loop dynamics. It also, on a statistical basis, forces the PN System out of sync in accordance with the frequency established during our Phase I study (see section 3.5.1.2, Volume II).

A detailed description of the modified programs can be found in Appendices C-3&4. A high level input/output interface diagram for DLL is shown in Figure 5.6. Subroutine PNMON's interface diagram remains unchanged from that shown in the Phase I report.



*Inputs From Other Programs

†Read From Data Cards

EXTERNAL PARAMETERS†

FIGURE 5.6. INPUT/OUTPUT PARAMETERS FOR SUBROUTINE DLL.

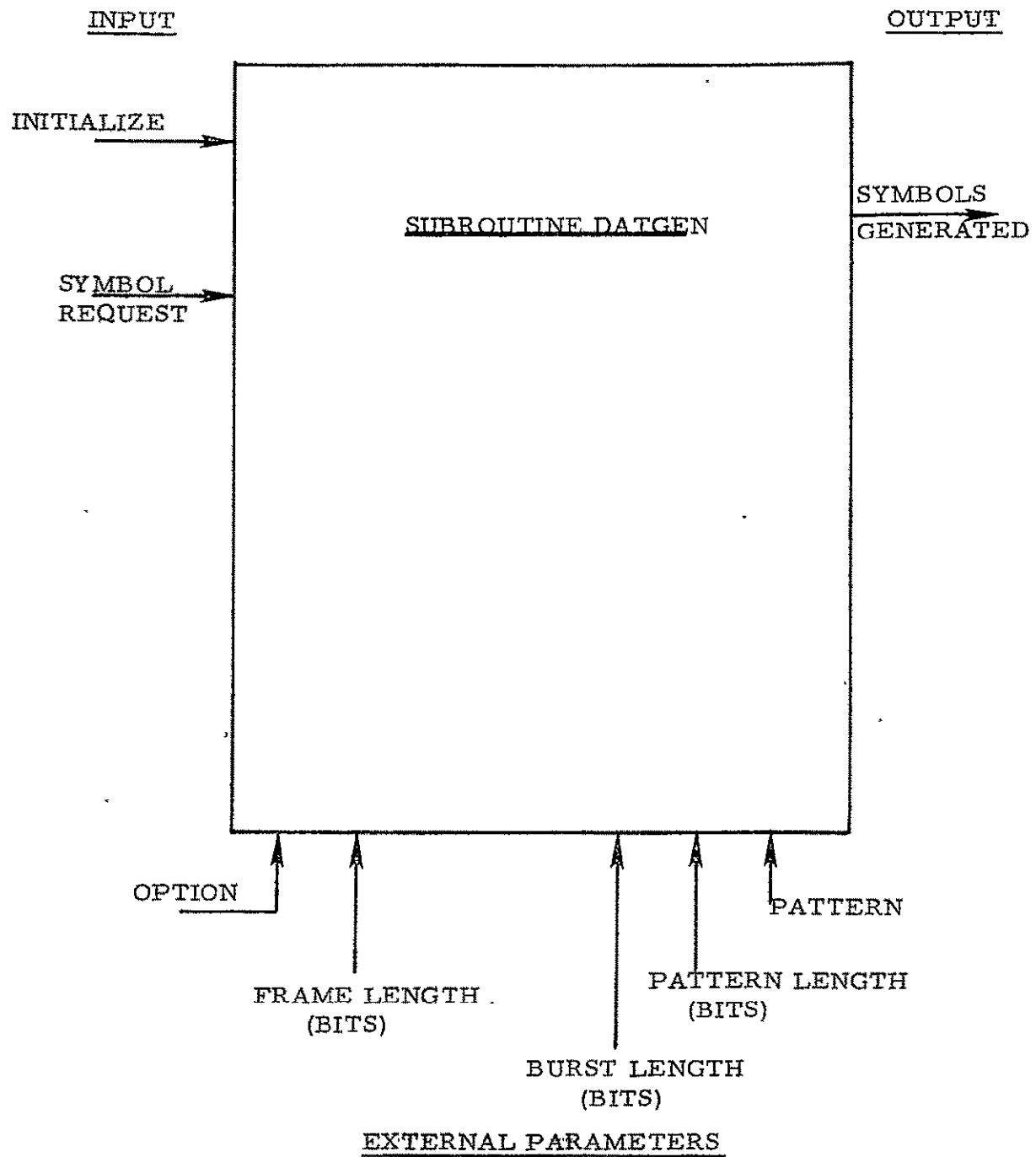


Figure 5.8. INPUT/OUTPUT PARAMETERS FOR SUBROUTINE DATGEN.

5.2.2 Subroutine RFSS

The Manchester encoding of randomly generated symbols and the doppler profile characteristics are simulated in RFSS with this subroutine obtaining the Gaussian input noise samples via calls to AANDN or RANDN. In the Phase I effort RFSS generated its own random symbol pattern. In Phase II RFSS has been modified to obtain its symbol source via calls to a new subroutine DATGEN. A detailed description of RFSS can be found in Appendix C-5 . The input/output parameters for RFSS are shown in Figure 5.7 .

5.2.3 Subroutine DATGEN

DATGEN supplies RFSS with a random, fixed, or random and fixed source of transmitted symbols. An input/output interface diagram is shown in Figure 5.8 . There are five parameters inputted via data cards. An option parameter determines whether the symbols generated are all random or are a combination of random and patterned values. The user selects and enters a 32 bit pattern, a bit at a time. The user also enters the number of bits in the pattern that are to be used. The total number of bits in the frame and the number of bits of the pattern transmitted each frame are also entered.

Figure 5.9 shows the frame organization. At the beginning of each frame a burst of M bits of the specified pattern $\{d_1, d_2, \dots, d_L, \dots, d_{32}\}$ is transmitted followed by N-M bits of random data to complete the frame. On the following frame the next M bits of pattern are used. The process of transmitting M bits of the L bit pattern continues until all bits are used and then the epoch will repeat. The bits $\{d_{L+1} \dots d_{32}\}$ although entered by the user are not used. Many possible patterns and harmonic relationships are possible with the format employed. A few of them are tabulated in Figure 5.9 . It is to be noted that a fixed pattern is obtained by equating M and N.

A detailed description of subroutine DATGEN can be found in Appendix C-6 .

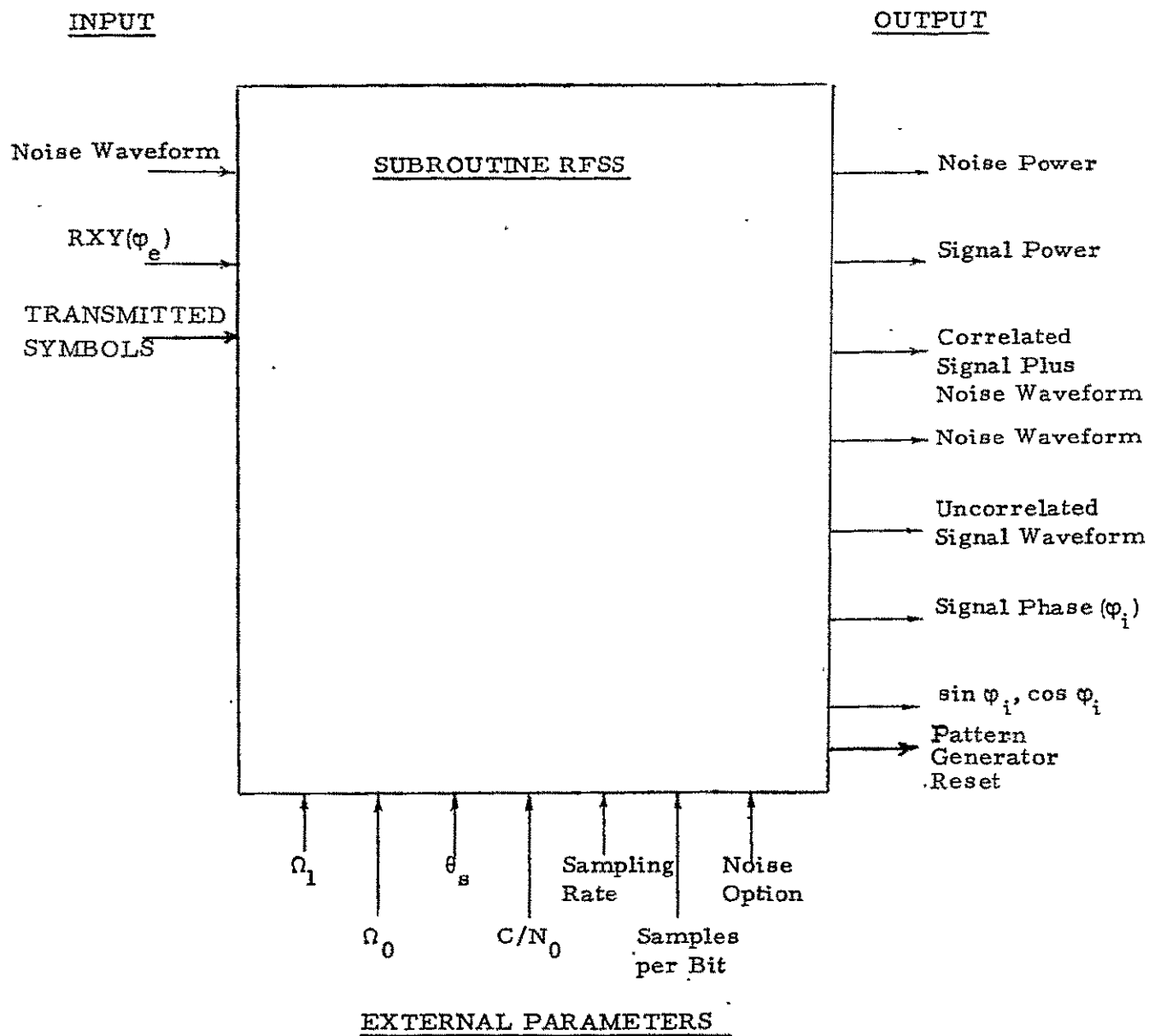
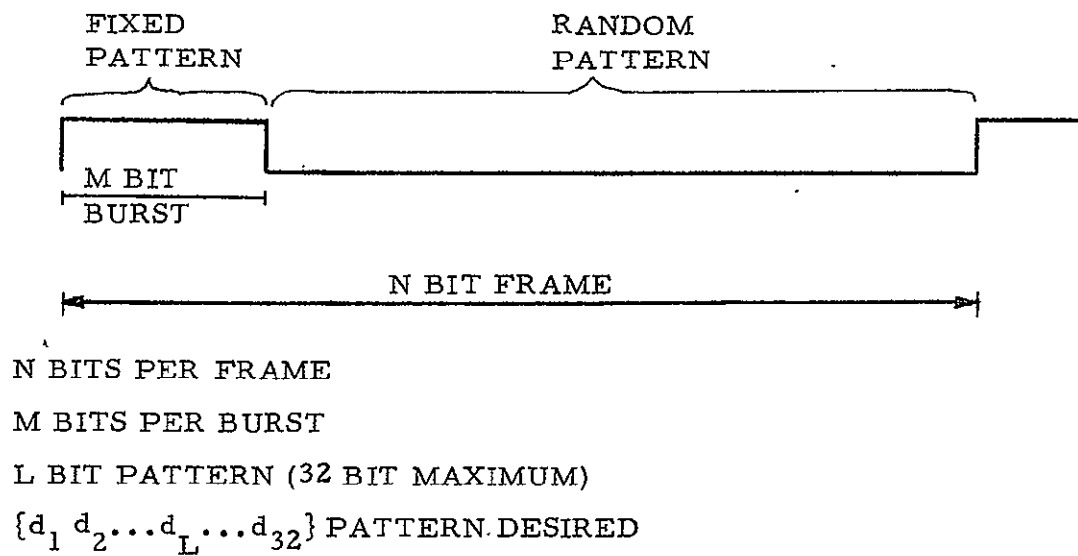


FIGURE 5.7. INPUT/OUTPUT PARAMETERS FOR SUBROUTINE RFSS.

FIGURE 5.9. FRAME ORGANIZATION.



PATTERN	N	M	COMMENT
$\{d_1 d_2 \dots d_L \dots d_{32}\}$	100	1	Single Bit Sync Pattern of the First L Bits
$\{d_1 d_2 \dots d_L \dots d_{32}\}$	100	2	Double Bit Sync Pattern of the First L Bits
$\{d_1 d_2 \dots d_{32}\}$	100	16	16 Bit Preamble
$\{d_1 \dots d_{32}\}$	16	16	Repeating 16 Bit Pattern
$\{1 \dots\}$	1	1	All 1's (L=1)
$\{1 0 \dots\}$	2	2	Alternating (1, 0) (L=2)
$\{1 1 0 0 \dots\}$	4	4	Alternating (1100)(L=4)

SECTION 6

SYSTEM STUDY

6.0 Introduction

This section serves to describe and document the results of experiments performed to evaluate overall behavior of the integrated system. One series of experiments measures the symbol error rate as a function of filters and loop interactions and establishes the basic baseline performance. Another group of experiments evaluates the effects of limiting, removing and unbalancing filters, bandwidth variations, and channel equalization on overall performance. A third series of experiments evaluate system behavior in the presence of signal transients. Additional experiments on PN/Costas loop sweep interactions and PN acquisition behavior in the presence of excessive amounts of code doppler will be described.

6.1 System Baseline Performance

The baseline parameters used in the simulation are listed in Table 6.1. A series of experiments were undertaken to isolate and associate the sources of degradation with the various subsystems. The two solid curves in each of Figures 6.1 through 6.4 represent highly magnified versions of the ideal symbol error rate performance (SER) expanded about the 51.3 and 48.3 dB-Hz design points. Both the RF and second IF filters were removed in taking the measurements associated with Figure 6.1. The measured performance with all loops operating, all loops perfectly aligned, and only selected loops operating are plotted in this figure. Near ideal behavior results when all loops are bypassed. With all loops operating there is a measured degradation of .2 dB and .3 dB at input C/N_0 values of 51.3 and 48.3 dB-Hz, respectively. The majority of this loss is associated with the delay lock loop.

TABLE 6.1. BASELINE PARAMETERS

RF FILTER - 4 POLE CHEBYSHEV

BANDWIDTH = 22 MHz

RIPPLE FACTOR = 1 dB

IF FILTER - 2 POLE BUTTERWORTH

BANDWIDTH = 864 KHz

SYMBOL RATE = 216 K

DATA PATTERN = +1 every 100 Symbols

τ - DITHER LOOP - 48.3 dB-Hz DESIGN POINT

BANDWIDTH = 12.5 Hz

DAMPING FACTOR = .707

IF FILTER BANDWIDTH = 550 KHz

COSTAS LOOP - 51.3 dB-Hz DESIGN POINT

ARM FILTERS = 324 KHz

BANDWIDTH = 500 Hz

DAMPING FACTOR = 1.

SYMBOL SYNCHRONIZER - 48.3 dB-Hz DESIGN POINT

BANDWIDTH = 30 Hz

DAMPING FACTOR = 1.

SAMPLES PER SYMBOL = 10

GRAPH PAPER GRAPH-C CONTROLS CORPORATION Buffalo, New York
SQUARED 10 X 10 TO THE INCH AS-0806 -01
Revised in U.S.A.

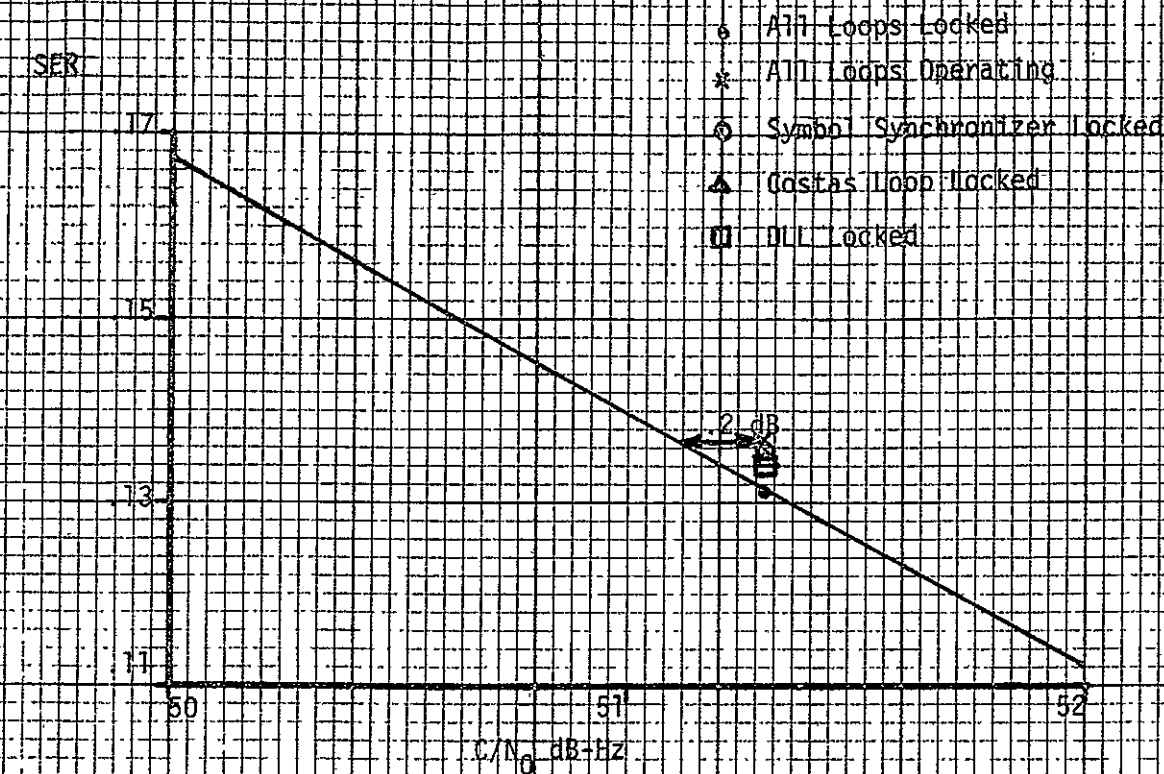


FIGURE 6.1, SER vs C/N_0 , NO RF OR IF FILTER

TABLE 6.1. BASELINE PARAMETERS

RF FILTER - 4 POLE CHEBYSHEV

BANDWIDTH = 22 MHz

RIPPLE FACTOR = 1 dB

IF FILTER - 2 POLE BUTTERWORTH

BANDWIDTH = 864 KHz

SYMBOL RATE = 216 K

DATA PATTERN = +1 every 100 Symbols

τ - DITHER LOOP - 48.3 dB-Hz DESIGN POINT

BANDWIDTH = 12.5 Hz

DAMPING FACTOR = .707

IF FILTER BANDWIDTH = 550 KHz

COSTAS LOOP - 51.3 dB-Hz DESIGN POINT

ARM FILTERS = 324 KHz

BANDWIDTH = 500 Hz

DAMPING FACTOR = 1.

SYMBOL SYNCHRONIZER - 48.3 dB-Hz DESIGN POINT

BANDWIDTH = 30 Hz

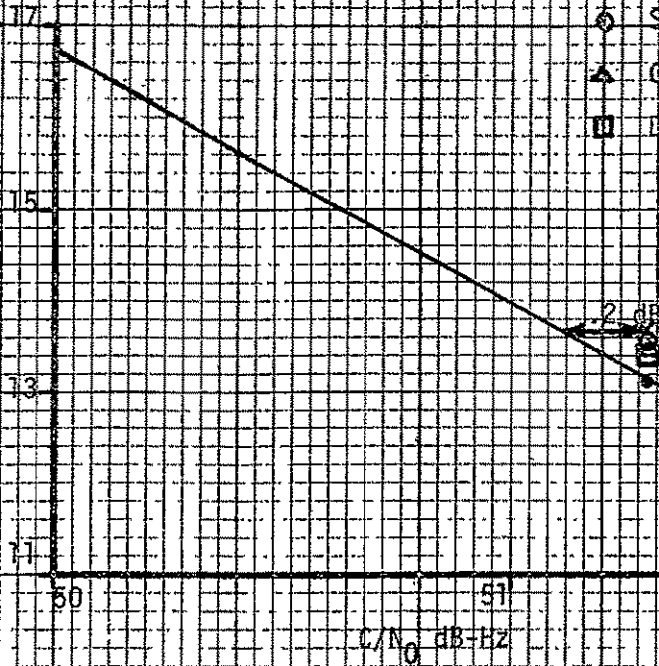
DAMPING FACTOR = 1.

SAMPLES PER SYMBOL = 10

LinCom

SER

- All Loops Locked
- * All Loops Operating
- ◊ Symbol Synchronizer Locked
- ▲ Costas Loop Locked
- All Locked



SER

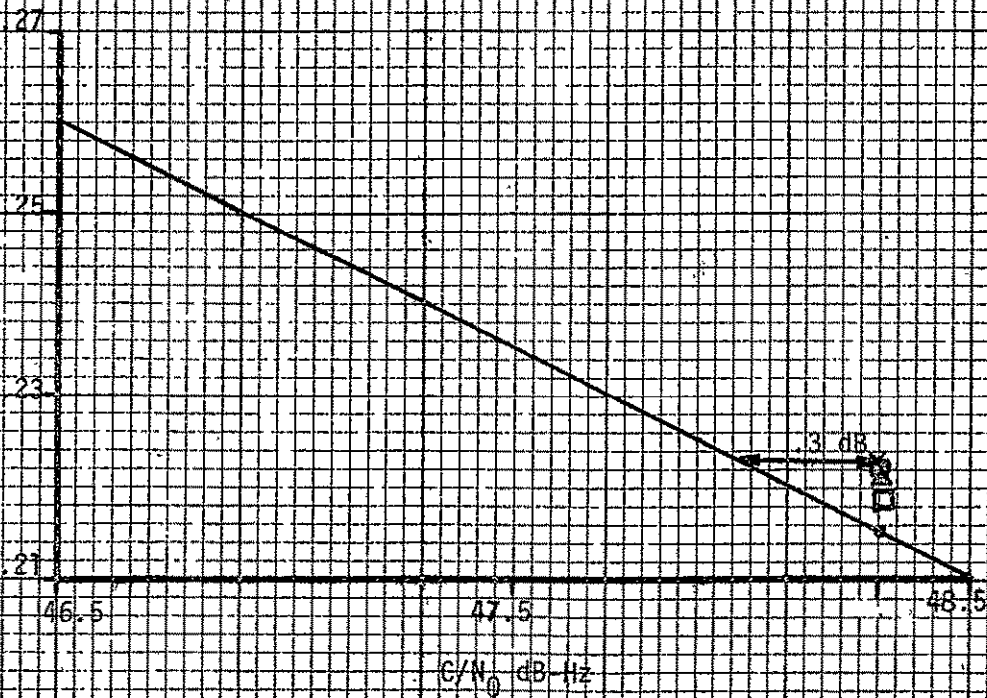


FIGURE 6.1, SER vs C/N_0 , NO RF OR IF FILTER

LinCom

[illegible]

FIGURE 6.2. SER vs C/N_0 , ALL LOOPS OPERATING

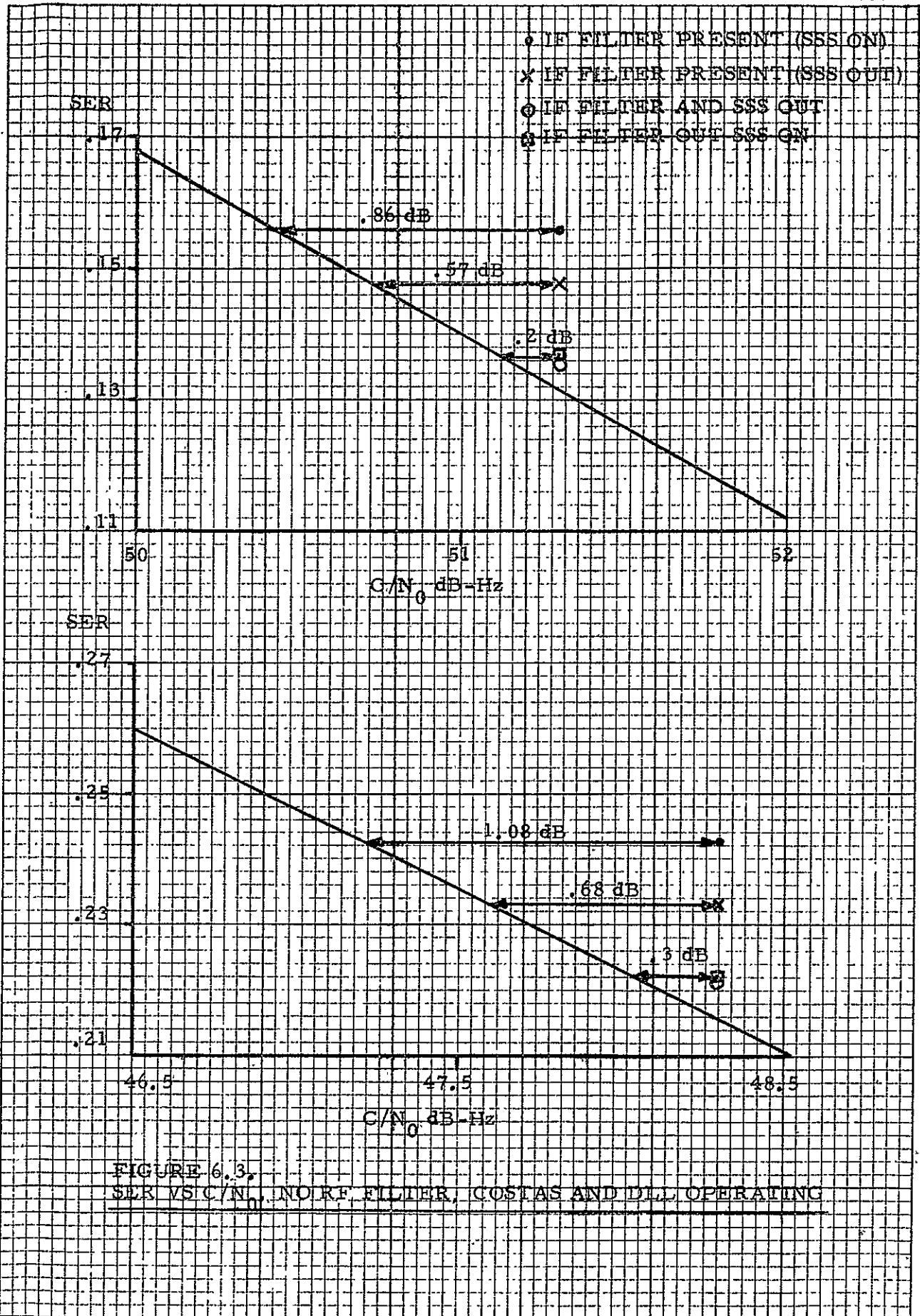


FIGURE 6.3, SER VS C/N_0 , NO RF FILTER, COSTAS AND DLL OPERATING

SQUARE 10 X 10 TO THE INCH AS 0486 -01

SEARCHED INDEXED SERIALIZED FILED
FEB 1964
COMMUNICATIONS CONTROL CORPORATION
BUFFALO NEW YORK
100-1052

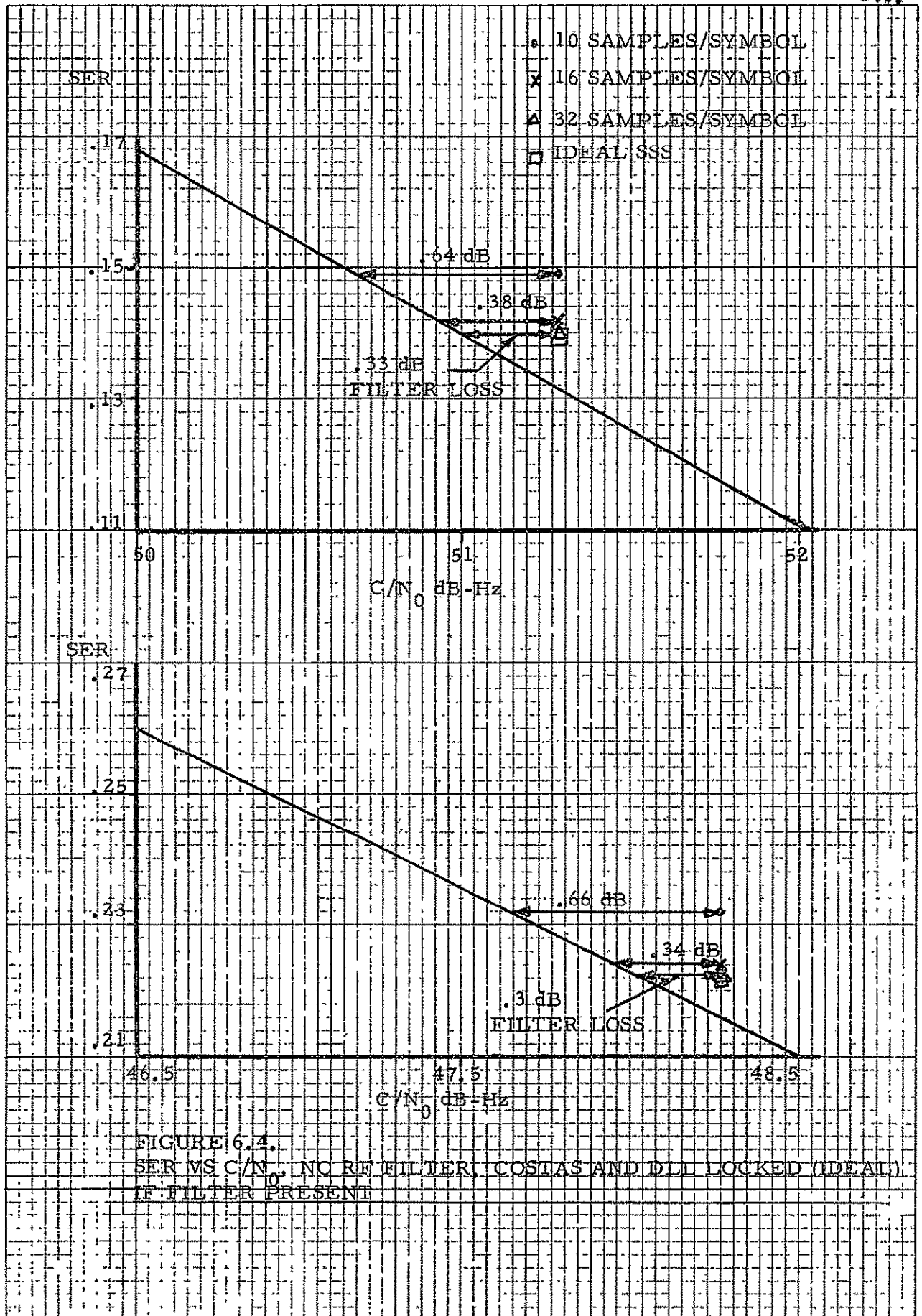


FIGURE 6.4.
SER VS C/N_0 , NO RF FILTER, COSTAS AND DLL LOCKED (IDEAL).
IF FILTER PRESENT

The effects of filtering on system performance can be seen from Figure 6.2 . As expected the addition of an RF filter degrades performance by about .35 dB. Adding both an RF and an IF filter ($BW = 4 \times 216000$ Hz) degrades performance by over 1 dB; about .7 dB of this being attributed to the IF filter. The source of this degradation can be determined from Figure 6.3. One sees that without an IF filter the symbol synchronizer introduces no measurable degradation. With an IF filter present and with the symbol synchronizer out (ideal operation) there is approximately a .35 dB loss, as expected, due to the filter by itself. An additional $\sim .3$ dB of loss is contributed by the symbol synchronizer in the presence of the IF filter. No such loss exists with this filter removed, however. The observed results makes sense from an engineering standpoint. The SSS performance is established using information obtained from around the symbol transition points. When the transition edges are not sharp but become rounded due to filtering a skew condition exists, causing a hunting type loop operation which, part of the time, produces a one sample offset in integrating the symbol. Three techniques become obvious for reducing this degradation. The first, of course, is not to use an IF filter. If this is not allowable, one can reduce the degradation to an acceptably small value by increasing the sampling rate as shown in Figure 6.4. A third technique, channel equalization, will be discussed in Section 6.3.5.

6.2 Measurement Accuracy

One notes some inconsistencies in measurement values (fractions of a tenth of a dB) when comparing the data in Figures 6.1 through 6.4 . It is appropriate, at this point, to caution the reader in attempting to evaluate degradations to within a fraction of a dB. Simulation results are measured values. Associated with any

measurement, whether it be related to hardware or software is a degree of uncertainty. Truly establishing measurement accuracies to within fractions of a tenth of a dB requires more samples, i.e. symbols detected, than were normally used. This statement is illustrated with the aid of Figure 6.5 . Hypothesize a system under test whose actual error rate is shown as the dot in the figure. If one were to measure the system performance, the measured SER would not necessarily correspond to the dot but would take on values falling, 90% of the time, within a range shown by the vertical line. The more samples used, the narrower this range. With 4000 symbols per measurement the predicted degradation will be within $\pm .25$ dB with a 90% confidence. For 25000 symbols, predicted degradation will be within $\pm .11$ dB, 90% of the time.

The exact number of symbols per measurement used in obtaining the data of Section 6.1 and other data to be subsequently discussed was variable, ranging between 4000 to 25000 symbols/test. Four thousand symbols per test were used when a relative performance measure was desired, i.e. change in performance with or without a filter, using identical noise samples for each run. Twenty-five thousand symbols were processed to obtain an absolute measure of performance, i.e. degradation from theoretical curve associated with the IF filter and symbol synchronizer. The relative measurement technique was used most frequently as it allowed us to perform many experiments with only minimum amounts of CPU time.

6.3 System Reconfiguration

The backbone configuration shown in Figure 6.6 displays the location of the principle mixers and filters used in simulating the spacecraft receiver. The shaded blocks were the subject of a series of experiments performed to evaluate the effects of limiting,

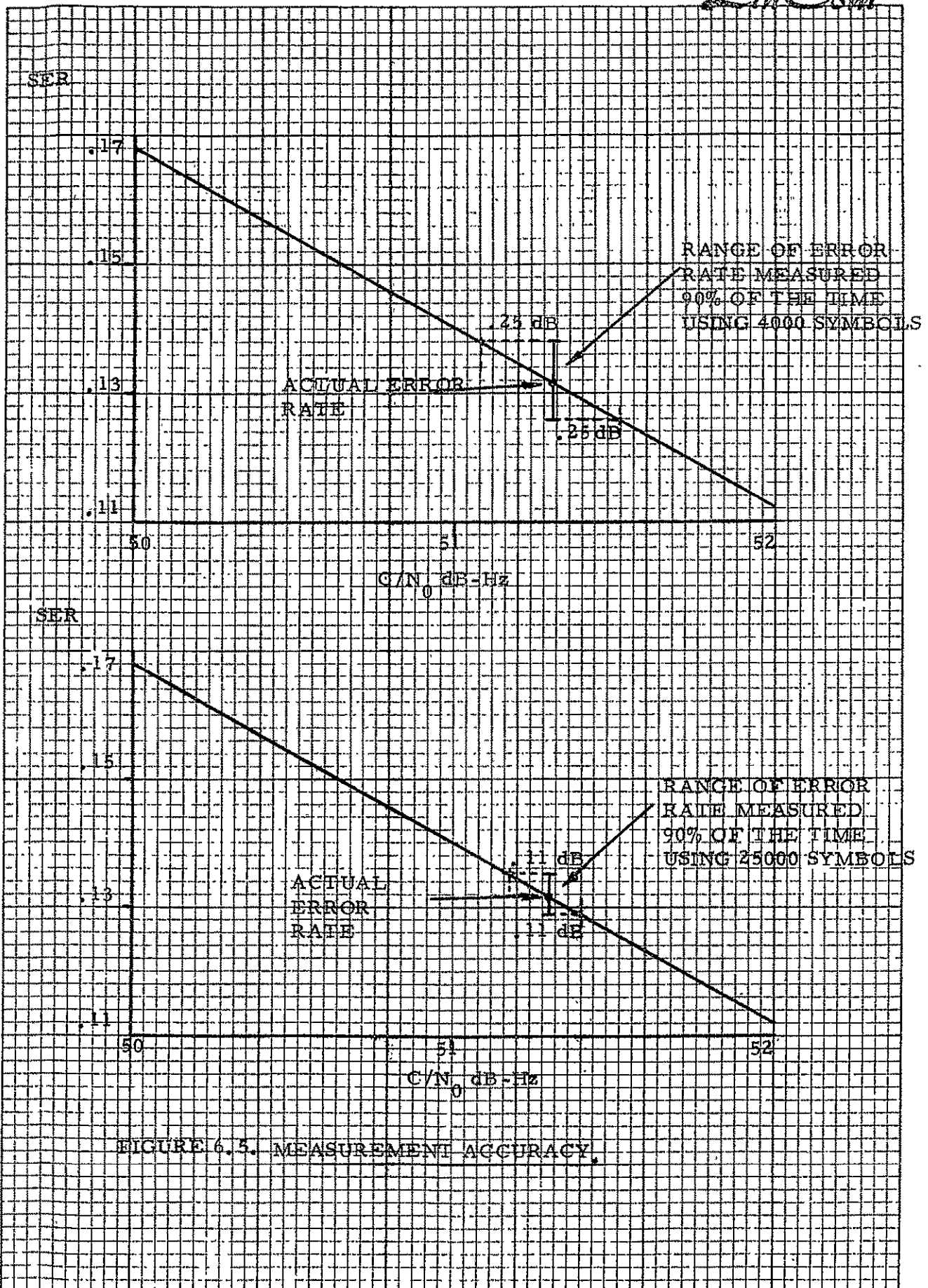


FIGURE 6.5. MEASUREMENT ACCURACY.

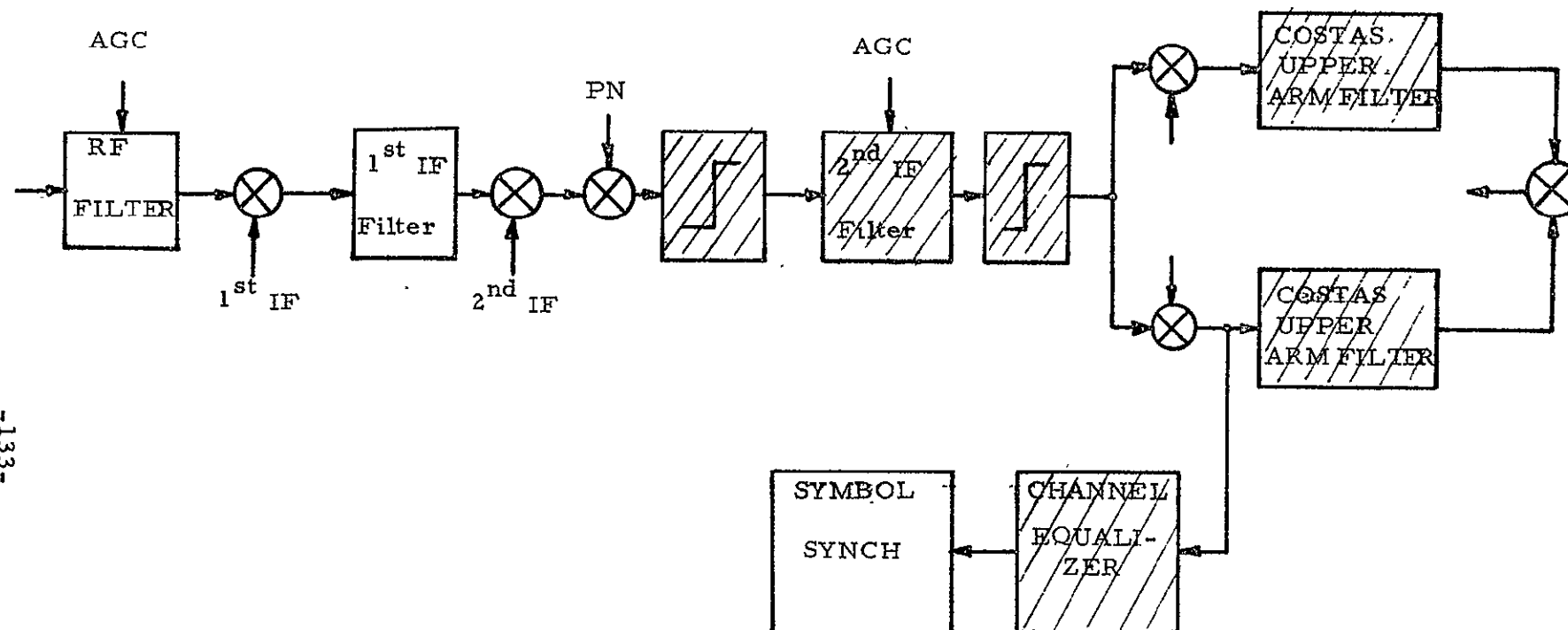


FIGURE 6.6. BACKBONE CONFIGURATION ILLUSTRATING PLACEMENT OF PRINCIPLE MIXERS AND FILTERS.

removing and unbalancing filters, bandwidth variations, and channel equalization on overall performance.

6.3.1 Costas Arm Filter Unbalanced

The results of an experiment evaluating the effects on SER performance when the Costas arm filters are unbalanced is shown in Figure 6.7 . It appears that an unbalance of 2/1 in arm filter bandwidths has no appreciable effect on the SER. The Costas loop jitter was also monitored while running the experiment. These measurements are tabulated in Table 6.2. Interestingly the minimum jitter condition occurs when the upper arm filter bandwidth is twice that of the lower arm filter. It can be concluded from this experiment that the tracking capabilities of our receiver is not sensitive to Costas arm filters. (At least for the design point parameters used.) Carrier acquisition properties were not evaluated as a function of filter unbalance.

6.3.2 Costas Arm Filters Removed

Since tracking performance is insensitive to arm filter imbalance (once again for our baseline parameter system) the immediate question is what happens if both arm filters are removed? The results of an experiment which attempts to answer this question are plotted in Figure 6.8. There is approximately .1 dB measured degradation with the arm filters removed. Note an increased phase jitter, however, particularly with the IF filter removed.

6.3.3 IF Filter Bandwidth Variations

Effects of varying the 2nd IF filters bandwidth can be observed from data displayed in Figure 6.9 . With the SSS present the optimum bandwidth is infinite, but if one is bandlimited it appears that there is a "best" bandwidth setting somewhat less than $4 R_s$.

GRAPH PAPER | GRAPHIC COMMUNICATIONS CORPORATION | Buffalo, New York
 SQUARE 10 X 10 TO THE INCH AS 0405 -01
 Printed in U.S.A.

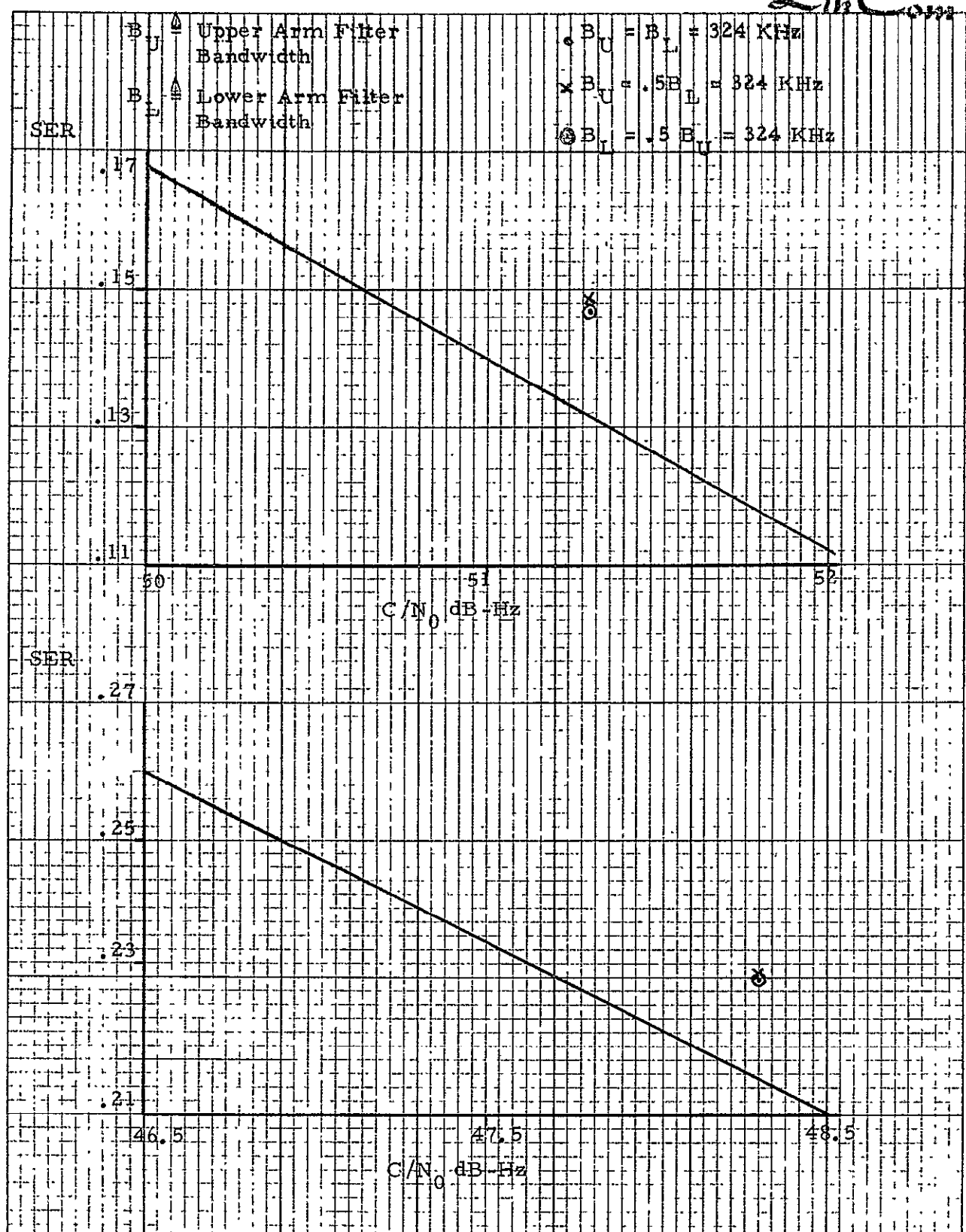


FIGURE 6.7.
 SER vs C/N_0 , RE FILTER, NO IF FILTER, LOOPS OPERATING

TABLE 6.2. COSTAS LOOP RMS JITTER (2ϕ) IN RADIANS AS A FUNCTION OF ARM FILTER PARAMETERS (NO IF FILTER)

MEASUREMENT	$C/N_0 = 51.3$			$C/N_0 = 48.3$		
	$B_U = 324K$ $B_L = 648K$	$B_U = 324K$ $B_L = 324K$	$B_U = 648K$ $B_L = 324K$	$B_U = 324K$ $B_L = 648K$	$B_U = 324K$ $B_L = 324K$	$B_U = 648K$ $B_L = 324K$
1	.2356	.202	.185	.383	.256	.23
2	.2782	.192	.15	.46	.31	.26
3	.244	.20	.16	.41	.32	.27
4	.2616	.21	.2	.4	.34	.28
5	<u>.2552</u>	<u>.212</u>	<u>.19</u>	<u>.38</u>	<u>.31</u>	<u>.3</u>
AVERAGE	.255	.203	.177	.406	.307	.268

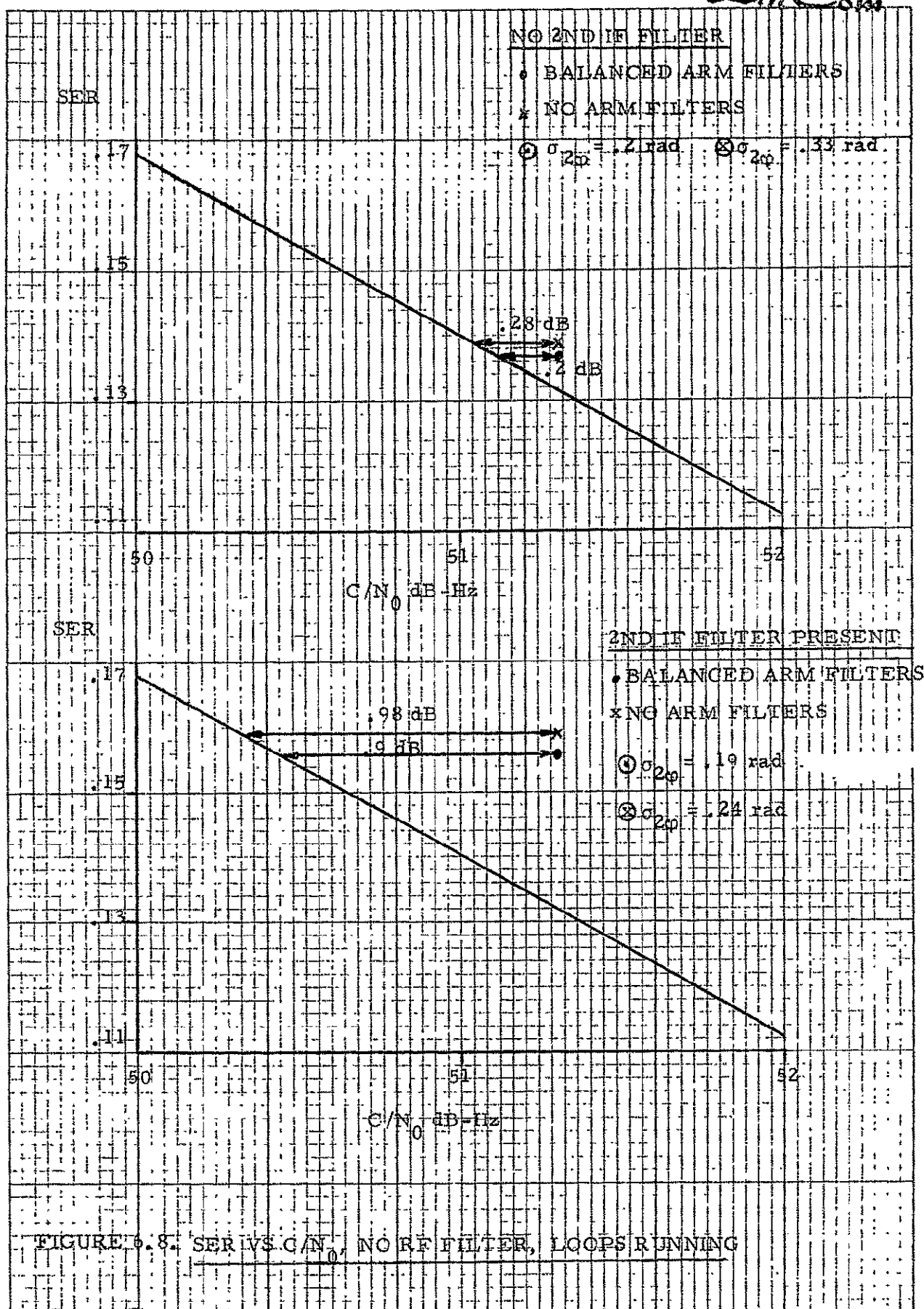
LOOPS RUNNING

NO IF FILTER

RF FILTER PRESENT

LinCom

GRAPHIC COMMUNICATIONS, 8444 Ave 127, 1
SQUARE 10 X 10 TO THE INCH AS-8005 -01
SERIES, U.S.A.



LinCom

LinCom

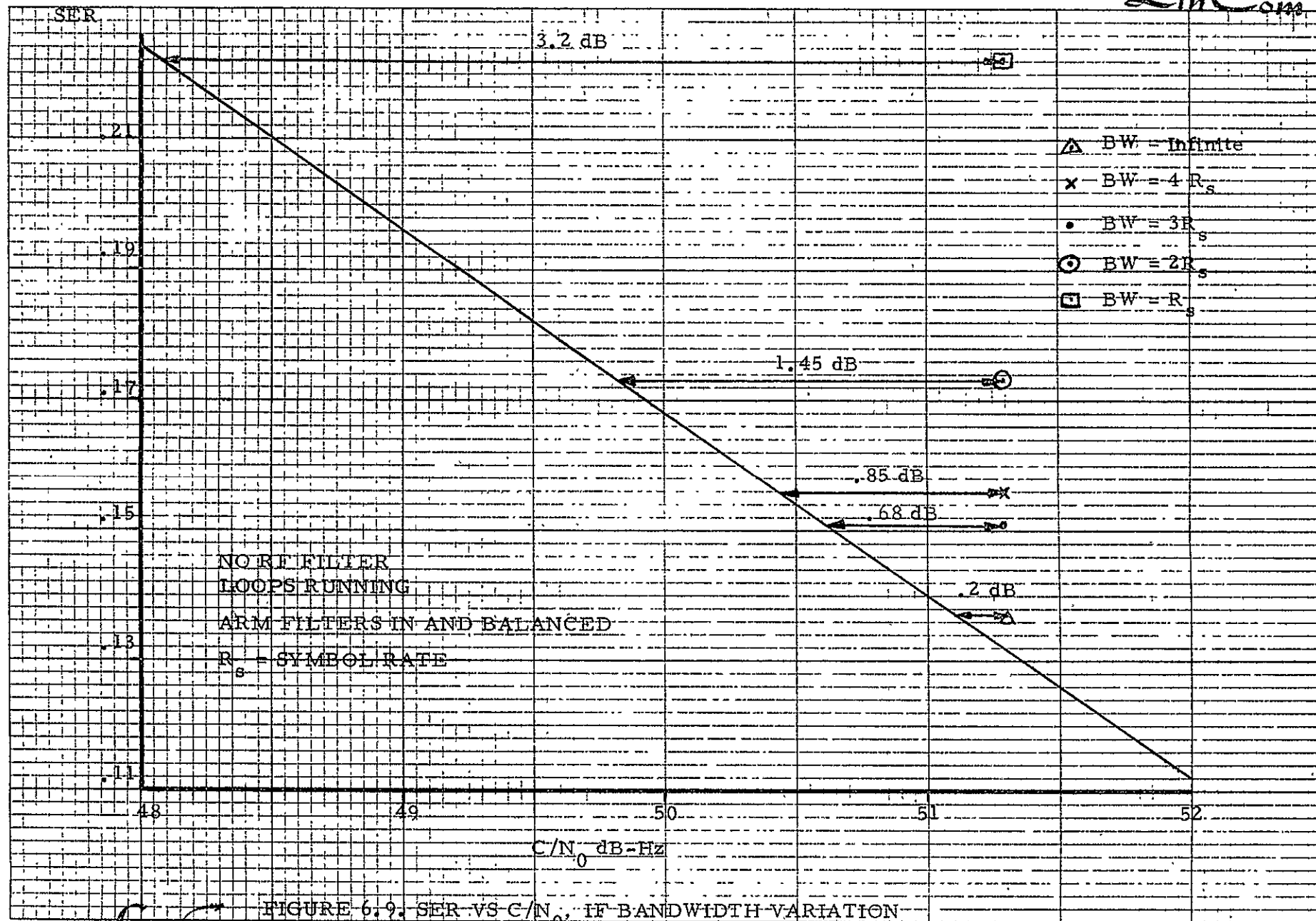


FIGURE 6.9. SER VS C/N_0 IF BANDWIDTH VARIATION

LinCom

6.3.4 Limiting

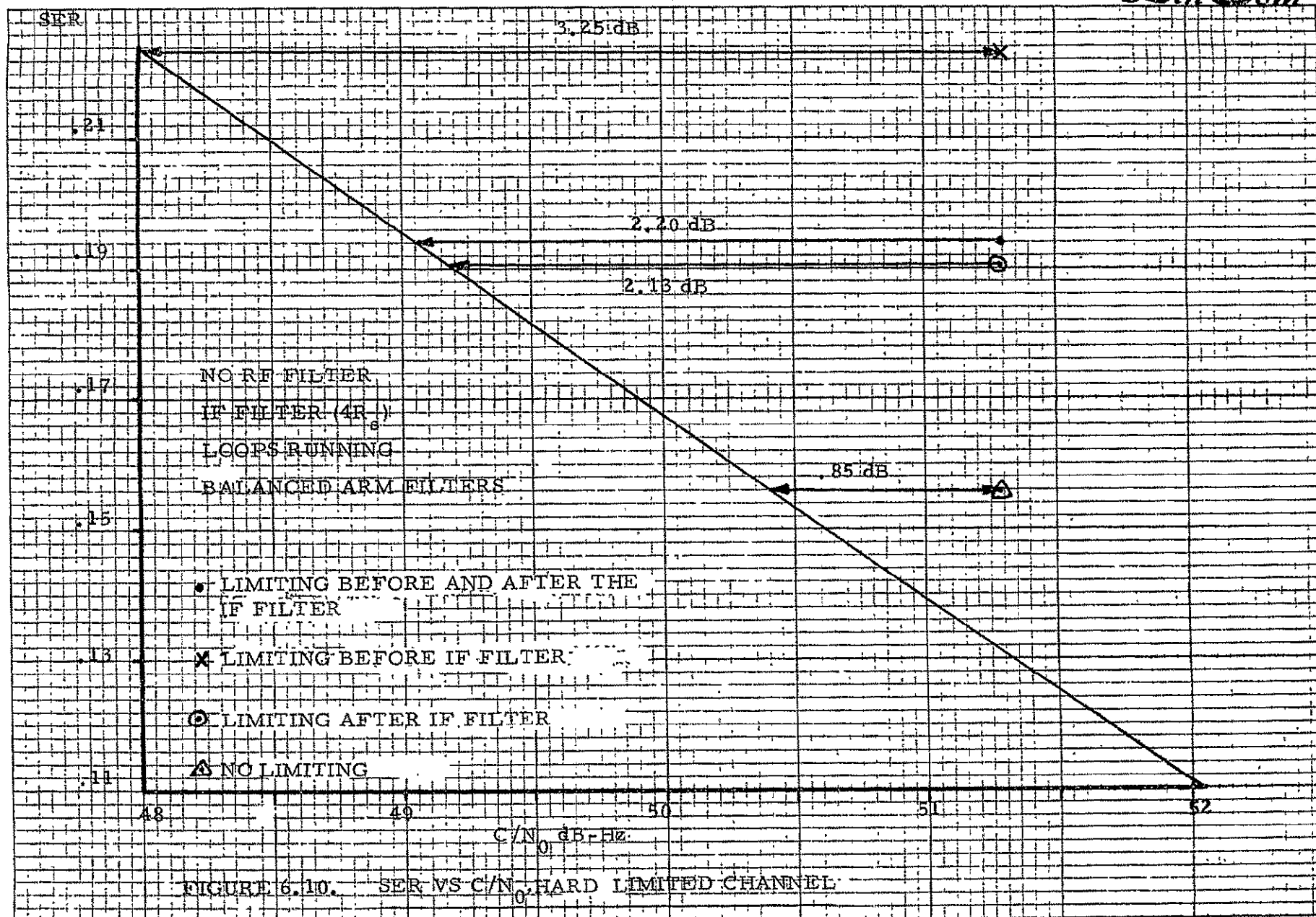
A simple experiment was performed to evaluate the impact, on system tracking performance, of a hardlimited channel. The results are plotted in Figure 6.10. As shown in Figure 6.6 a limiter could be placed before, after, and before and after the 2nd IF filter. Serious degradation occurs for all conditions in our deep noise environment. Placing the limiter after the 2nd IF filter produces the least degradation (1.3 dB). One should not of course place the clipper before the filter (2.4 dB loss). If for some reason hard limiting does or needs to occur somewhere in the RF section, it might be possible to reduce its effect by adding a second limiter after the IF filter.

6.3.5 Channel Equalization

The following discussion is applicable to the system tracking behavior (SER and jitter at 51.3 dB-Hz). Suppose from a hardware consideration one wished to operate without arm filters. Figure 6.8 indicates that a 2nd IF filter would be desirable for jitter reduction. Yet by using such a filter system performance would degrade by about .6 dB. About .3 dB of this can be removed via an increase in SSS sampling rate, leaving a net loss of .3 dB. Channel equalization will not only reduce this degradation but also the one associated with SSS sampling rate.

The placement of the channel equalizer is shown in Figure 6.6. This filter has an inverse characteristic to that of the lowpass equivalent of the IF filter. The waveforms of Figure 6.11 illustrates the equalizers impact on signal reconstruction while data plotted in Figure 6.12 demonstrates its positive impact on system performance. Hardware implementation of this filter is quite simple since a digital symbol synchronizer is used. The equalizer with very little additional digital logic can be included as part of the digital symbol synchronizer.

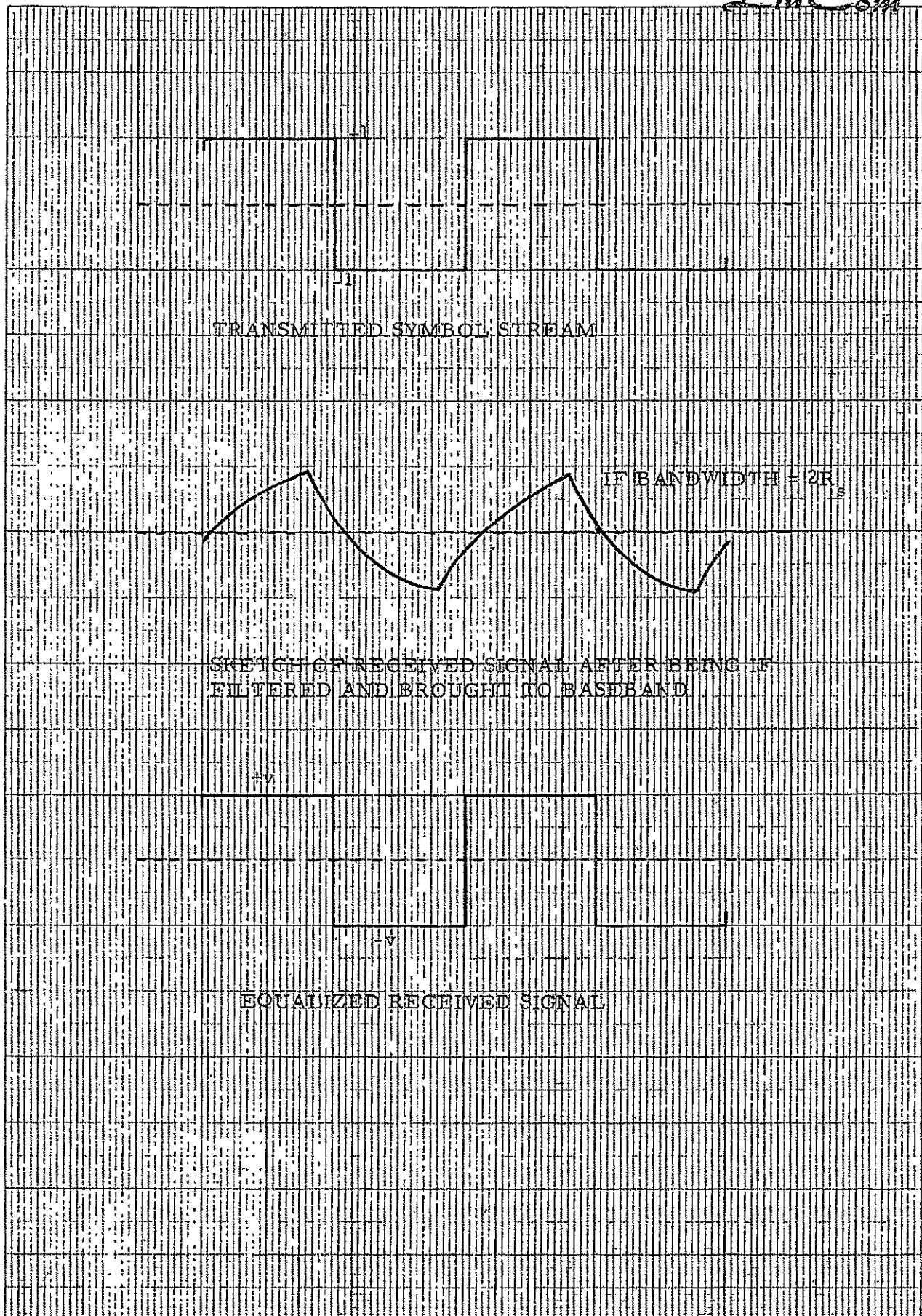
LinCom



LinCom

EUGENE DIETZGEN CO.
MADE IN U.S.A.

Q 340-20 DIETZGEN GRAPH PAPER
20 X 20 PER INCH



LinCom **FIGURE 6.11 CHANNEL EQUALIZATION WAVEFORMS.**

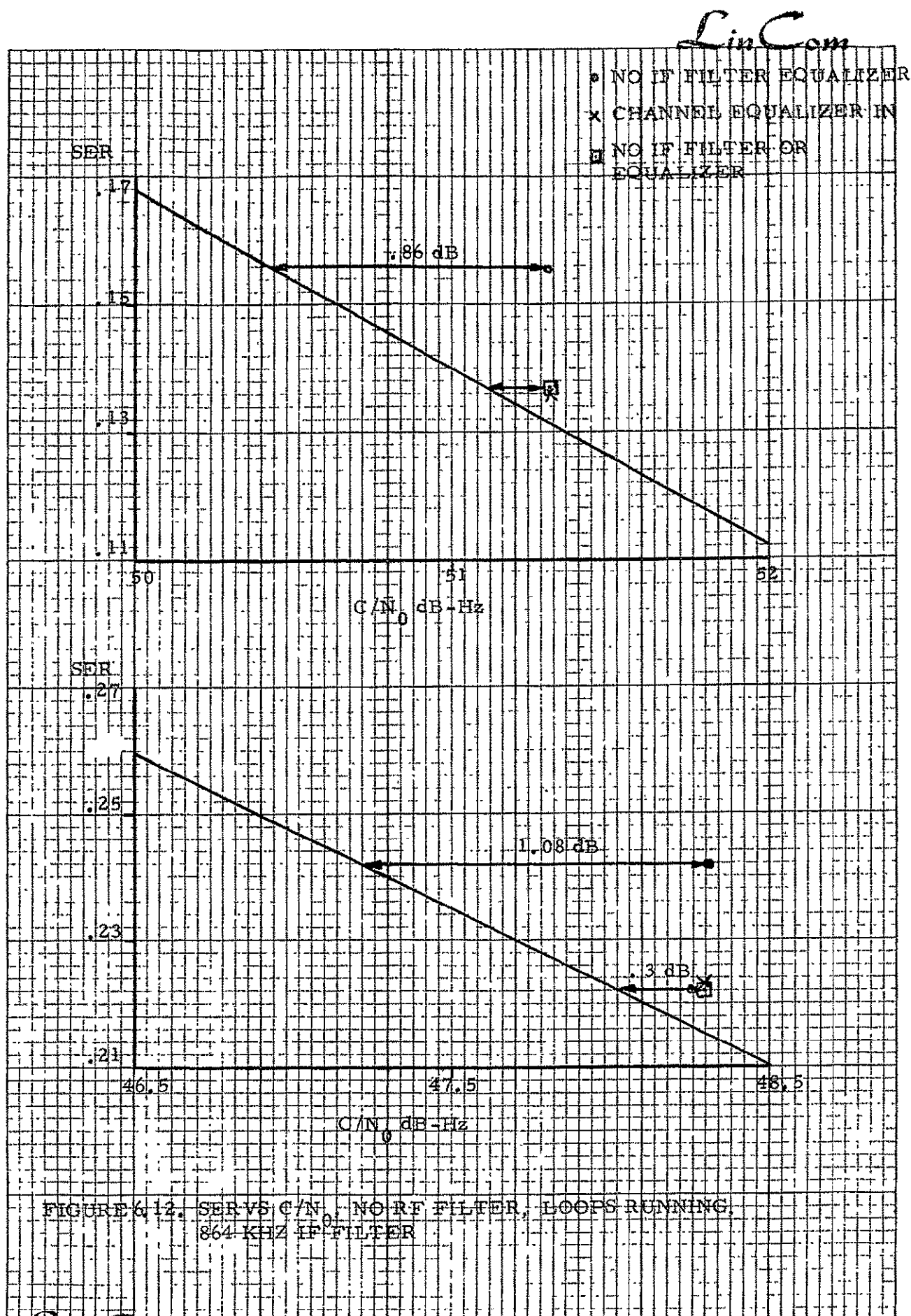


FIGURE 6-12. SERVS C/N: NO RF FILTER, LOOPS RUNNING
864 KHZ IF FILTER

6.4 System Response to Signal Transients

The normal sequence of events as the system (simulation) turns on, acquires, and enters the tracking mode will be described with the time-line waveform of Figure 6.13A . The simulation is activated at time T_0 with a C/N_0 value of 48.3 dB-Hz. The PN monitor logic immediately starts searching for code sync. The local code phase is stepped to within near alignment at T_1 , 13.745 seconds after T_0 . By T_2 , .010 seconds later, PN synchronization has been accomplished. At this time the microprocessor starts narrowing the τ -dither loop bandwidth and initiates carrier sweep. The local VCO is swept to near alignment with the carrier frequency at T_3 , 4.09 seconds later. At T_4 the VCO has locked to the incoming carrier. This locked condition is verified at time T_5 . At this time the system enters the tracking mode and data detection commences. For this trial, it then takes the system 17.855 seconds from start up to when a first symbol is processed.

One notes that in the above description the Symbol Synchronizer Subsystem was not mentioned. This was so, since there is no requirement for evaluating SSS acquisition performance. However, in passing, it should be pointed out that the SSS has been programmed for acquisition with a B_L of 1000 Hz which is narrowed to 30 Hz upon entering the tracking mode. For simplicity our SSS is activated with no noise as part of program initialization. It could just as easily have been activated at T_5 in which case our total acquisition time would increase approximately .01 seconds.

The results of a study evaluating system behavior during signal fades and jump discontinuities in carrier phase will be presented in the following sections. It is assumed as shown in Figure 6.13B that the perturbation starts some time, T_S , after the system has been tracking and lasts until time T_E at which point the normal operating condition is restored; the outage duration being T_I .

LinCom

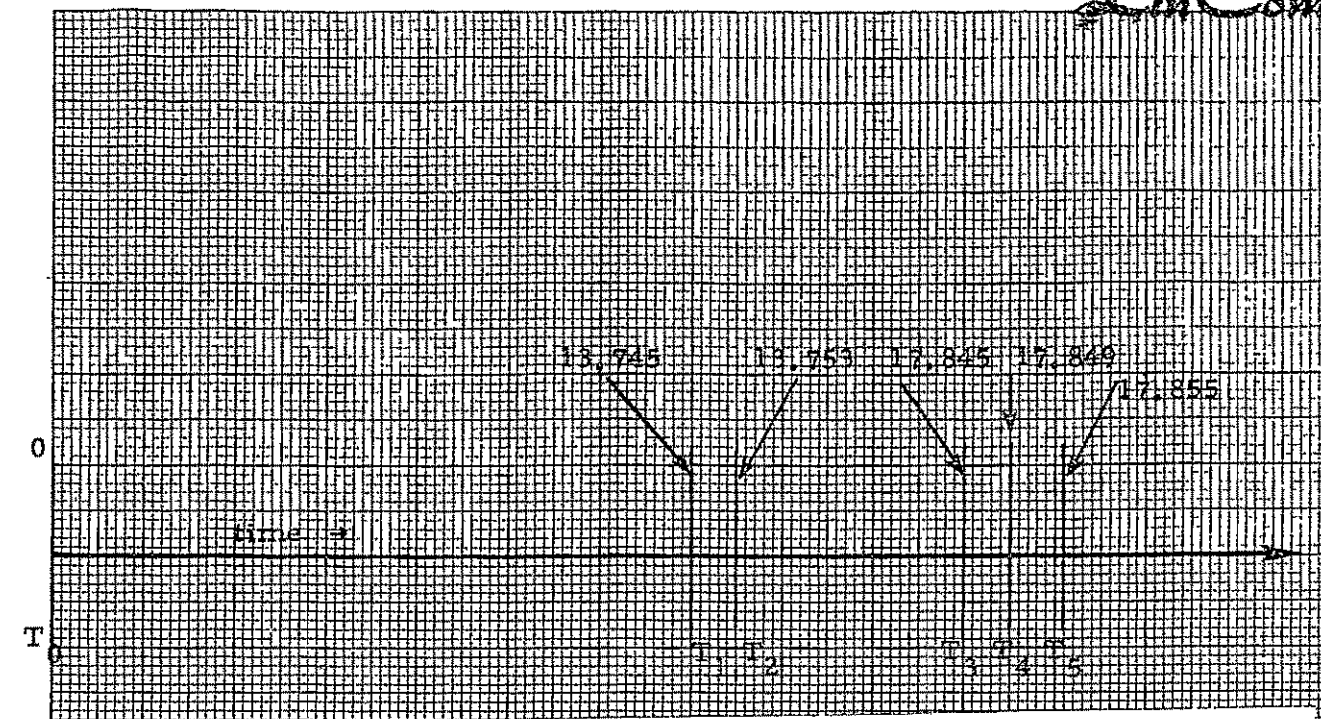


Figure 6.13a. TIME LINE WAVEFORM (NOT TO SCALE).

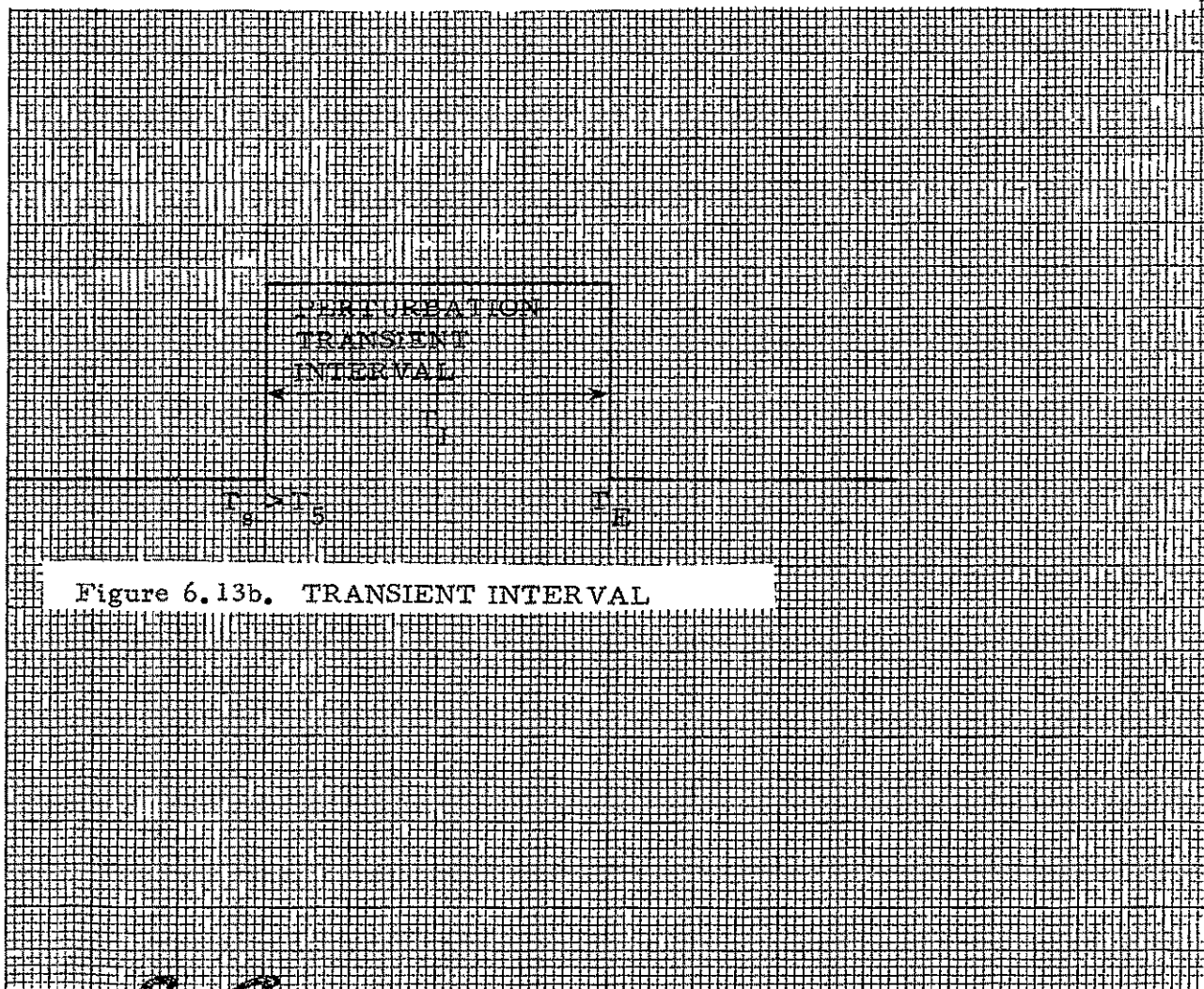


Figure 6.13b. TRANSIENT INTERVAL

LinCom

6.4.1 Signal Fades

If the duration of T_I is measured in seconds the system operation will, of course, collapse. The carrier recovery hardware will go first shortly followed by the PN despreaders. Of concern to us here, is system behavior for small values of T_I . The following discussion is based upon a series of computer experiments and analytical extrapolation at a C/N_0 of 48.3 dB-Hz.

As T_I is increased from zero the first noticeable effect would be an increase in symbol error rate to the 50% level until T_I approaches 12 milliseconds. The Carrier Recovery Subsystem will collapse within ± 2 milliseconds of this value with about a 90% confidence. Carrier sweep will then be initiated. If the signal was turned on after carrier sync was lost but prior to the PN despreaders losing sync it could take from a few milliseconds up to 8.6 seconds for carrier reacquisition to be once again possible. A few milliseconds of sweep will quickly move the VCO away from the carrier frequency, sweep direction determine 8.6 seconds or few millisecond time.

The PN acquisition hardware will declare itself out of sync about 82% of the time when T_I extends to 72 milliseconds. The exact point where this occurs is statistical but in greater than 94% of the cases the despreaders will lose sync if T_I extends to 194 milliseconds. The above sequence of events is summarized in Figure 6.14. Once the despreaders lose sync and starts its search the signal can remain out for seconds without having any further impact (~ 4.7 seconds for high C/N_0 and no code doppler, ~ 80 seconds for 48.3 dB-Hz C/N_0 and 300 code chips of doppler).

The above discussion centered around sync detector hardware and logic. In doing so, it was tacitly assumed that the timing loops introduced no further complications. The extent to which these

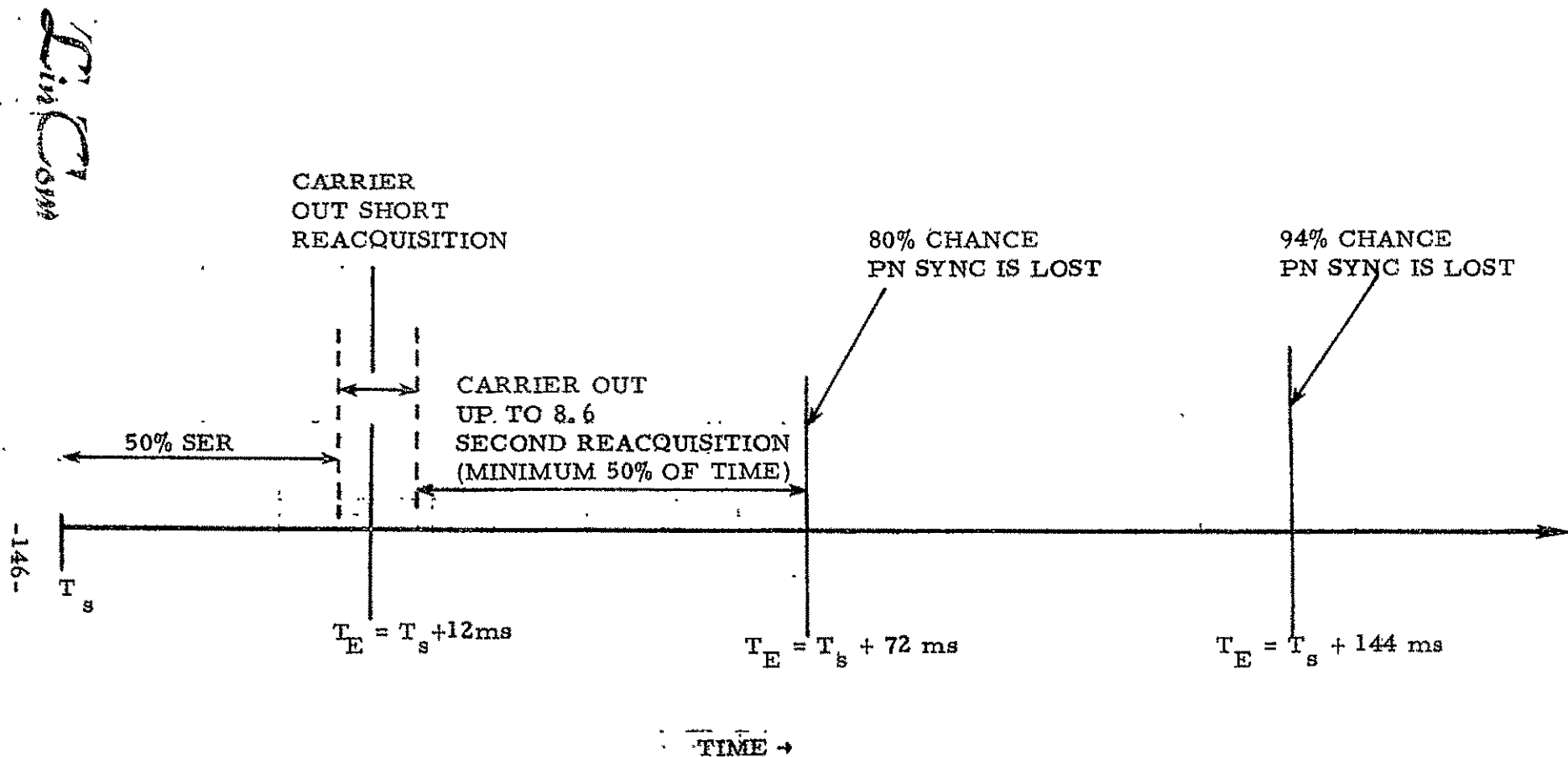


Figure 6.14. EVENT TIME LINE DURING SIGNAL OUTAGE.

loops effect system behavior during fades depends upon how well the charge (estimate of doppler offset) is held by loop filter capacitors. Ideal integrate plus proportional loop filters were used in the simulation with a discharge rate established by computer roundoff accuracy. The Costas loop ($B_L = 500$ Hz) will continue to track doppler for over 200 milliseconds with no noticeable problem. If the signal returns anywhere within this time interval the loop will quickly start tracking with nothing more than a few cycle slips. The code loop is even better with its 12.5 Hz bandwidth. Code doppler continues to be tracked seconds after the signal is removed. Thus as far as the simulation is concerned and for well designed hardware (low leakage op-amps, etc.) sync detector hardware and logic establishes system performance during signal fades.

6.4.2 Carrier Phase Transients

The software while operating in the track mode was subjected to a series of abrupt carrier phase discontinuities at a C/N_0 value of 51.3 dB-Hz. No noticeable effect occurred with the PN Subsystem. Effects on the Carrier Recovery Subsystem were not severe. Symbol error rate increased and there were a few cases of cycle slippage and one or two missed carrier sync decisions but no loss of the carrier was encountered. The major problem with phase jumps appears to be in the symbol error rate monitor software whenever the Costas loop slipped such that an inversion in the detected data occurred. The monitor software was self-correcting of these inversions within a 1000 symbol interval.

6.5 PN/Carrier Sweep Interaction

A study delving into potential problems associated with PN and carrier sweep interaction will be discussed in this section. A coherent transponder requirement constrains us to the long

loop carrier recovery implementation shown in Figure 6.15 .

Eighty MHz has been selected for both the VCO rest frequency and the 2nd IF filter's center frequency. The 2nd IF translation frequency is at 320 MHz (N=4). The 1st IF translation frequency is 1.6 GHz (M=5) and the 1st IF filters center frequency is at 400 MHz. It is assumed that the RF carrier is nominally set at 2 GHz. The local VCO is swept so as to produce a ± 44 KHz variation of the 1.6 GHz 1st IF translation frequency. This range covers the maximum expected doppler uncertainty ($F_{\max} = \pm 55$ KHz). A 40 KHz/second sweep rate is used. While sweeping, the frequency of the incoming signal will be offset from the IF filters center frequency position in the manner shown in Figure 6.15 . The worst case frequency offset seen by the 1st IF filter during the course of a sweep is + or -99 KHz. For the same worst case doppler conditions the 2nd IF filter sees a + or -107.8 KHz offset.

Injecting signals through filters with relative frequency offsets that are large with respect to their bandwidths introduces system degradations such as energy loss, intersymbol interference, sweep rate beat notes, etc. Negligible signal distortion will be introduced by the 1st IF filter since its bandwidth is large as compared to the chip rate. Measurable signal distortion will be introduced by the 2nd IF filters however, since here bandwidths are on the order of the symbol rate. It is to be noted that signal degradations will only exist when the carrier is being swept or is not in lock (locked the worst case 2nd IF filter offset is only 2.2 KHz).

The following hardware may be affected by the carrier sweep via this 2nd IF filter distortion:

- Automatic Gain Control
- PN Acquisition Circuits and Logic
- τ -Dither Loop
- Carrier Recovery Subsystem.

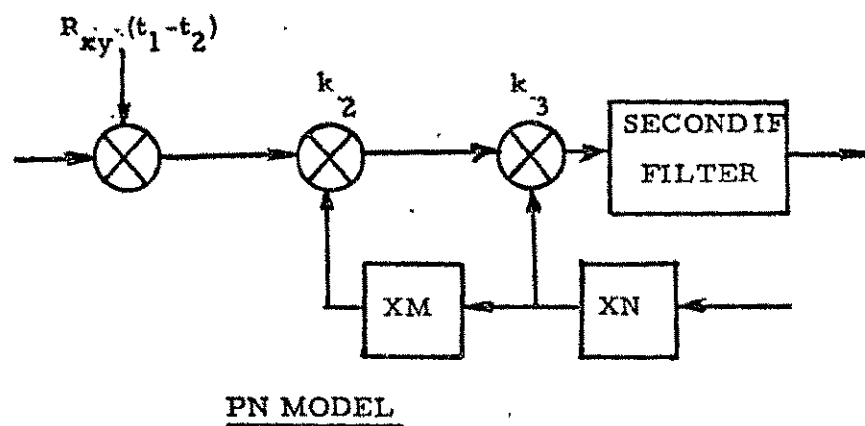
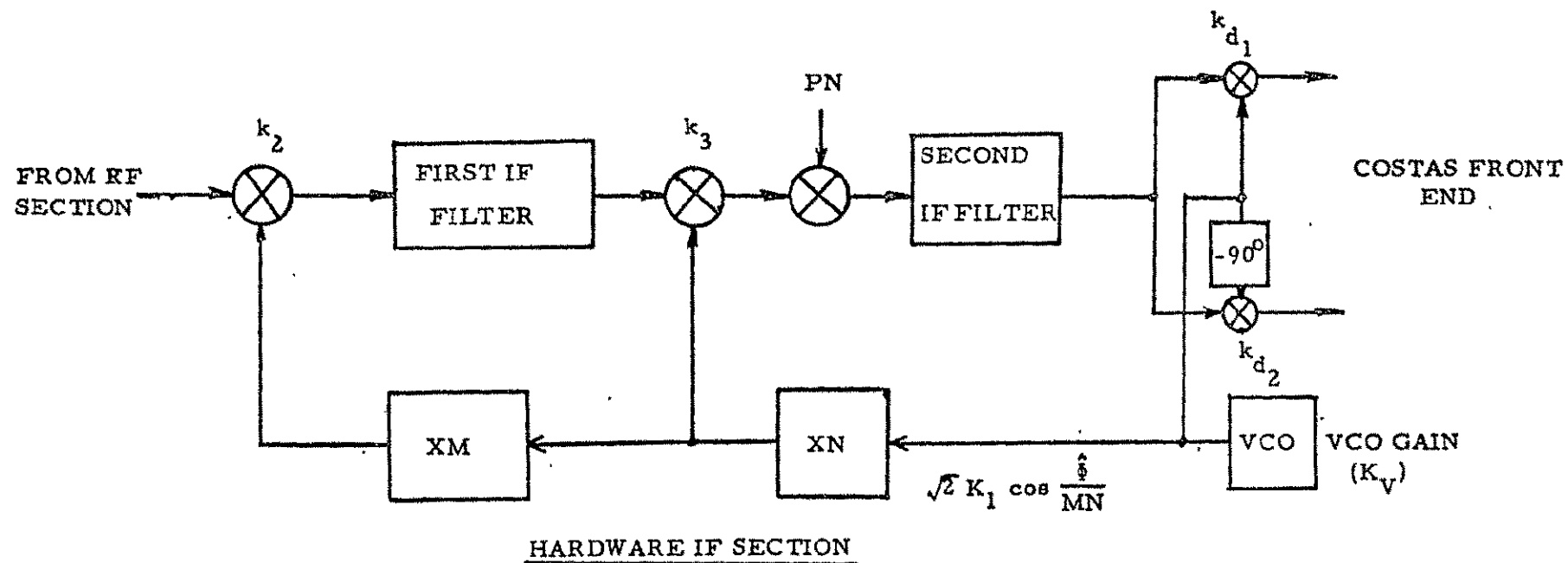
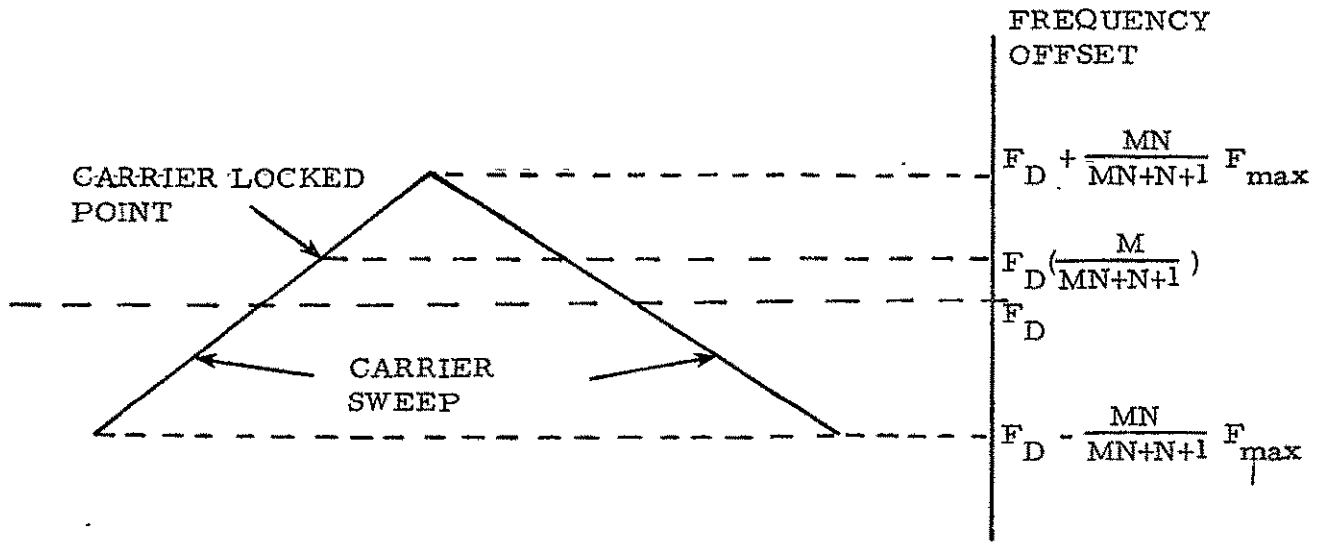
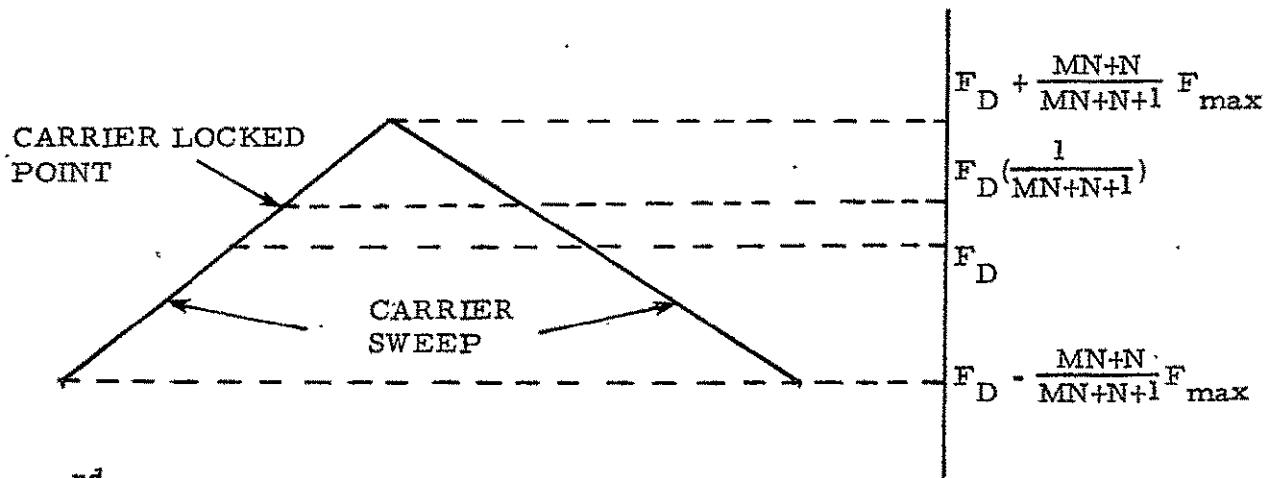


FIGURE 6.15. IF SECTION MODELING.



A. 1st IF FILTER FREQUENCY OFFSET



B. 2nd IF FILTER FREQUENCY OFFSET

F_{max} = magnitude of maximum doppler

F_D = actual doppler present

$M = 5, N = 4, MN+N+1 = 25$

Figure 6.16. IF FREQUENCY OFFSETS DURING CARRIER SWEEP.

Assuming the PN Subsystem is operating ideally, sweeping the VCO has no direct measurable impact on the Carrier Recovery Subsystem and its associated AGC. This is due to the fact that the Costas loop bandwidth is narrow and will not allow the loop to pull in until the VCO is in near perfect alignment, for this condition the 2nd IF filter frequency offset is very small. The above statement has been verified countless times via normal exercising of the computer simulation.

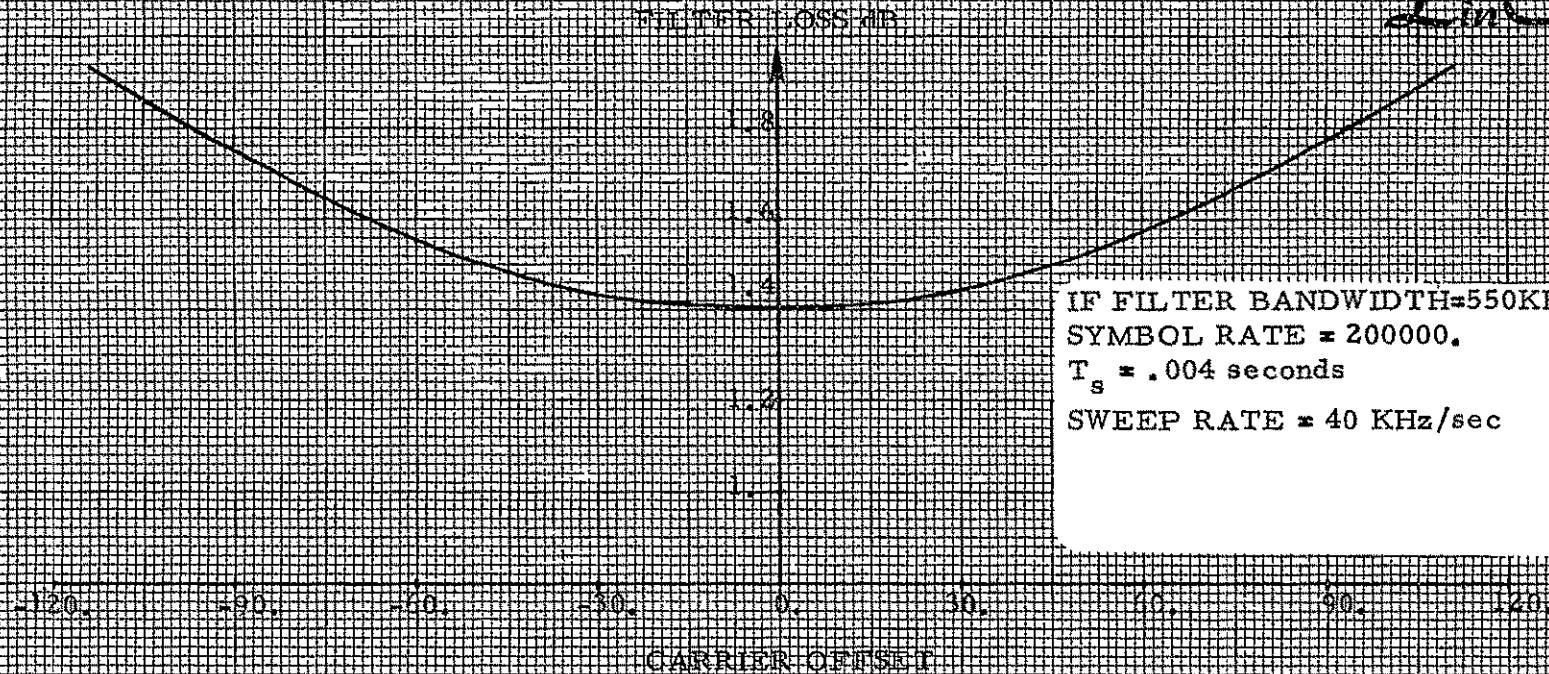
What has not, however, to date been studied is:

1. Possibility of the sweep forcing the PN Subsystem monitor logic to declare itself out of sync via a direct degradation of the sync detectors correlation voltage or degradation of the noise estimate.
2. Possibility of the sweep causing the PN Subsystem to be thrown out of sync via degradations to our narrowband code tracking loop.
3. Possibility of the sweep degrading the correlation voltage sufficiently to not allow the Carrier Recovery Subsystem to lock.

A series of experiments, requiring extensive software modification, were performed to evaluate these and search out other potential problem areas.

The first experiment involved sweeping the VCO and evaluating the signal at the output of the PN sync detectors integrate and dump circuit. Results are presented in Figure 6.17 for a signal only condition. Over the full range of a sweep the additional degradation is never more than .5 dB. This small additional loss will not significantly affect sync decisions when one considers that the PN system has already acquired the code with approximately a 2.5 dB penalty and is now in track where a rather stiff fly wheel action is operational. With the signal removed and noise only present the

LinCom



CARRIER OFFSET MEASURED RELATIVE TO CENTER FREQUENCY
OF IF FILTER
FILTER LOSS MEASURED RELATIVE TO THE NO FILTER CASE AT
OUTPUT OF PN SYNC DETECTOR

LinCom Fig. 6-17. FILTER LOSS VS CARRIER OFFSET.

output signal levels with and without sweeping are identical as can be seen from Table 6.3 . This experiment serves to verify that possibility #1 above will not occur. Repeating the experiment with signal and noise (48.3 dB-Hz) also supports this conclusion.

In a second experiment the complete program was exercised with and without doppler and sweep. Tests points were appropriately placed so as to monitor PN sync integrate and dump voltage, the correlation voltage driving the Costas loop, and code phase error. The results, shown in Figure 6.18, serve to verify that possibility #2 will not occur. This data was taken without noise and the program was activated for a period of time that was large compared to the DLL response time. No problems were observed when the experiment was repeated with signal and noise (48.3dB-Hz). In fact, no problems were encountered in any of the experiments until the sweep rate was increased to 2 MHz/second. Some anomalies then started to appear that may be either software or hardware in origin.

Insensitivity of the PN Subsystem to the sweep, particularly as reflected by Figure 6.18B , indicates that possibility #3 will not occur.

6.6 PN Acquisition-Large Code Doppler Offsets

The LinCom despreaders was designed to operate with code doppler uncertainties within a ± 300 code chip per second range. A question was raised at the Phase II PDR as to how the system behaved with doppler stress outside this range. The origin of this question was related to a problem encountered with TRW's hardware when the code doppler approached ± 400 code chips/second. From a limited understanding of the problem, their system as doppler increased, reached a point where acquisition was no longer possible. A series of experiments were performed to determine whether the simulation had a similar problem and discover its causes.

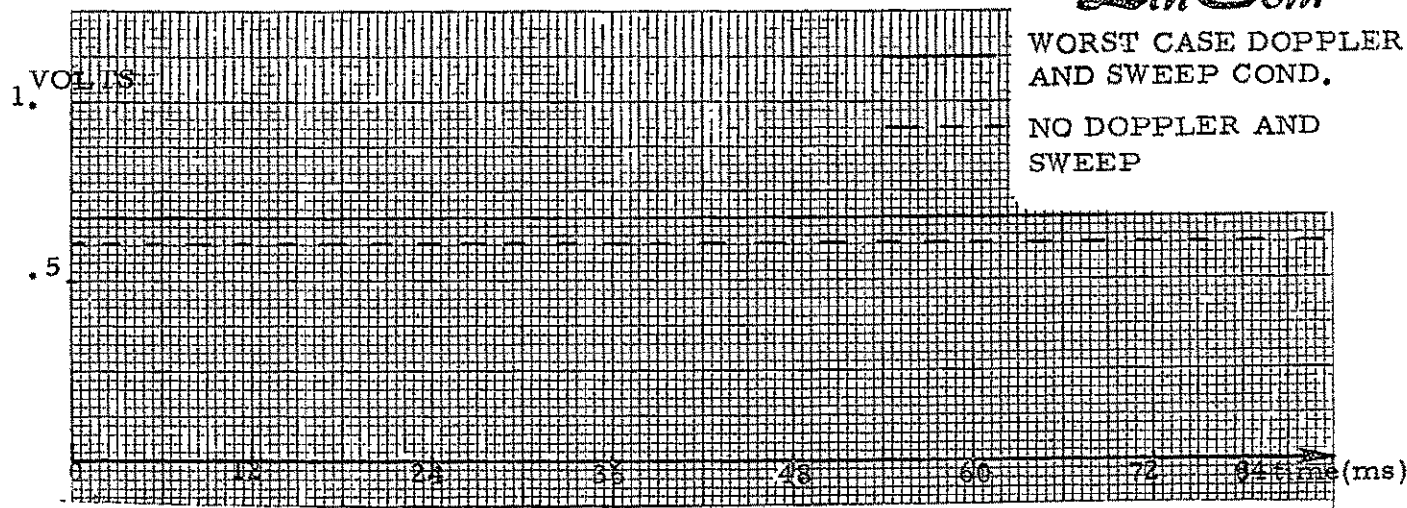
The first experiment involved activating the total despreaders package and evaluating the mean number of code epochs searched until

TABLE 6.3. SAMPLES AT OUTPUT OF PN SYNC DETECTOR FOR IDENTICAL CONDITIONS SHOWN IN FIGURE 6.17 WITH ONLY NOISE PRESENT

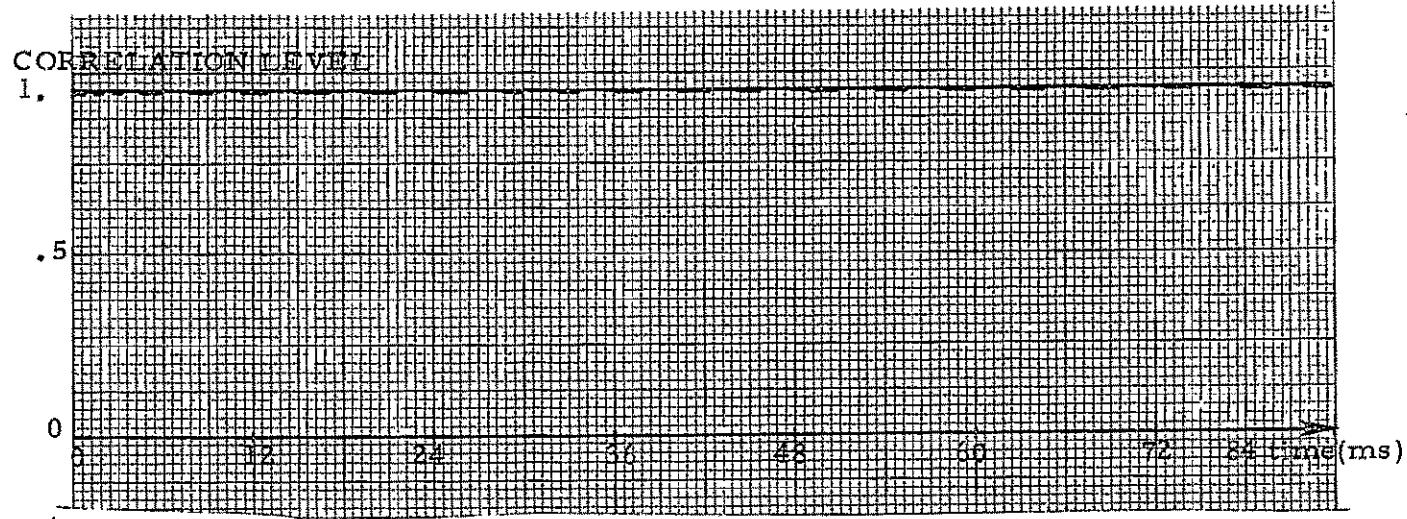
Time (Seconds)	No Doppler No Sweep	Initial Freq = 110 KHz Sweep = 40 KHz
.004	1.052	1.052
.008	1.075	1.075
.012	1.019	1.019
.016	1.045	1.045
.020	1.094	1.094
.024	1.052	1.052
.028	1.060	1.060
.032	1.090	1.090
.036	1.069	1.069
.040	1.068	1.068
.044	1.073	1.073
.048	1.025	1.025
.052	1.064	1.064
.056	1.031	1.031
.060	1.046	1.046
.064	1.073	1.073
.068	1.057	1.057
.072	1.034	1.034
.076	1.033	1.033
.080	1.039	1.039

Note: Identical Noise Samples were used in these two tests.

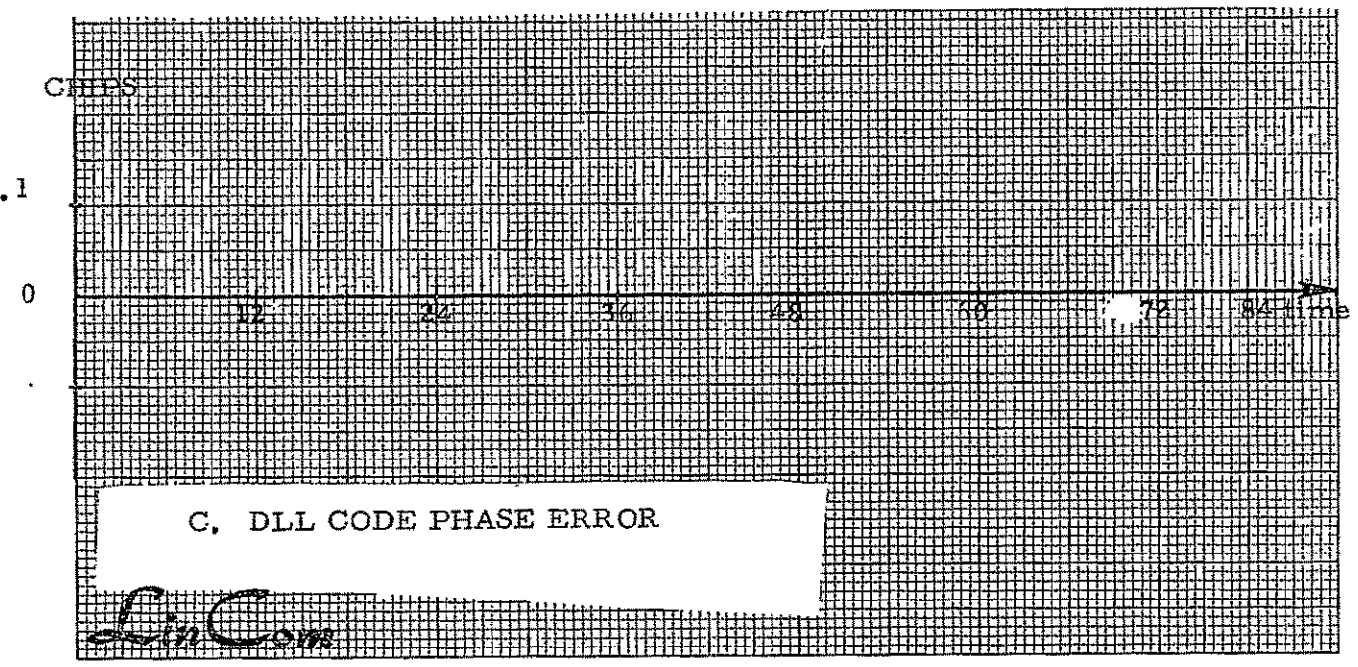
LinCom



A. PN SYNC DETECTOR I & D OUTPUT



B. CORRELATION LEVEL INJECTED INTO CARRIER SUBSYSTEM



LinCom

Figure 6.18. PN/CARRIER SWEEP INTERACTION.

46 1323

KEUFFEL & ESSER CO. MADE IN U.S.A.

sync occurs with both 300 and 400 chips/sec doppler present. Only qualitative results were obtained from this experiment due to execution time restrictions. In summary the mean number of code epochs were about the same for both values of doppler. At 400 chips/sec the amount of time the code dwelled near alignment significantly increased over that of the lower doppler value but eventually sync would occur. The mean time to search an epoch was also higher at 400 chips/sec doppler. It was necessary to partition the software and run experiments at the subsystem level to obtain quantitative results using reasonable amounts of execution time. These experimental results indicate that there may indeed be a problem caused by either:

1. Search strategies and parameters used or
2. τ-dither loop behavior and loop/search strategy interaction.

6.6.1 Doppler Limitation due to Search Strategies

Assuming an ideal delay locked loop acquisition time is given by

$$T_{acq} = \frac{2-p}{2p} T_e$$

where T_e is the effective time to completely cycle all code phase positions once in the absence of signal and p , a function of C/N_0 , doppler, and BT product used, is the probability of detecting sync during a single search epoch. The effective time may be expressed as

$$T_e = \frac{2 \times 2047 T_D}{1 - 2\Delta f T_D}$$

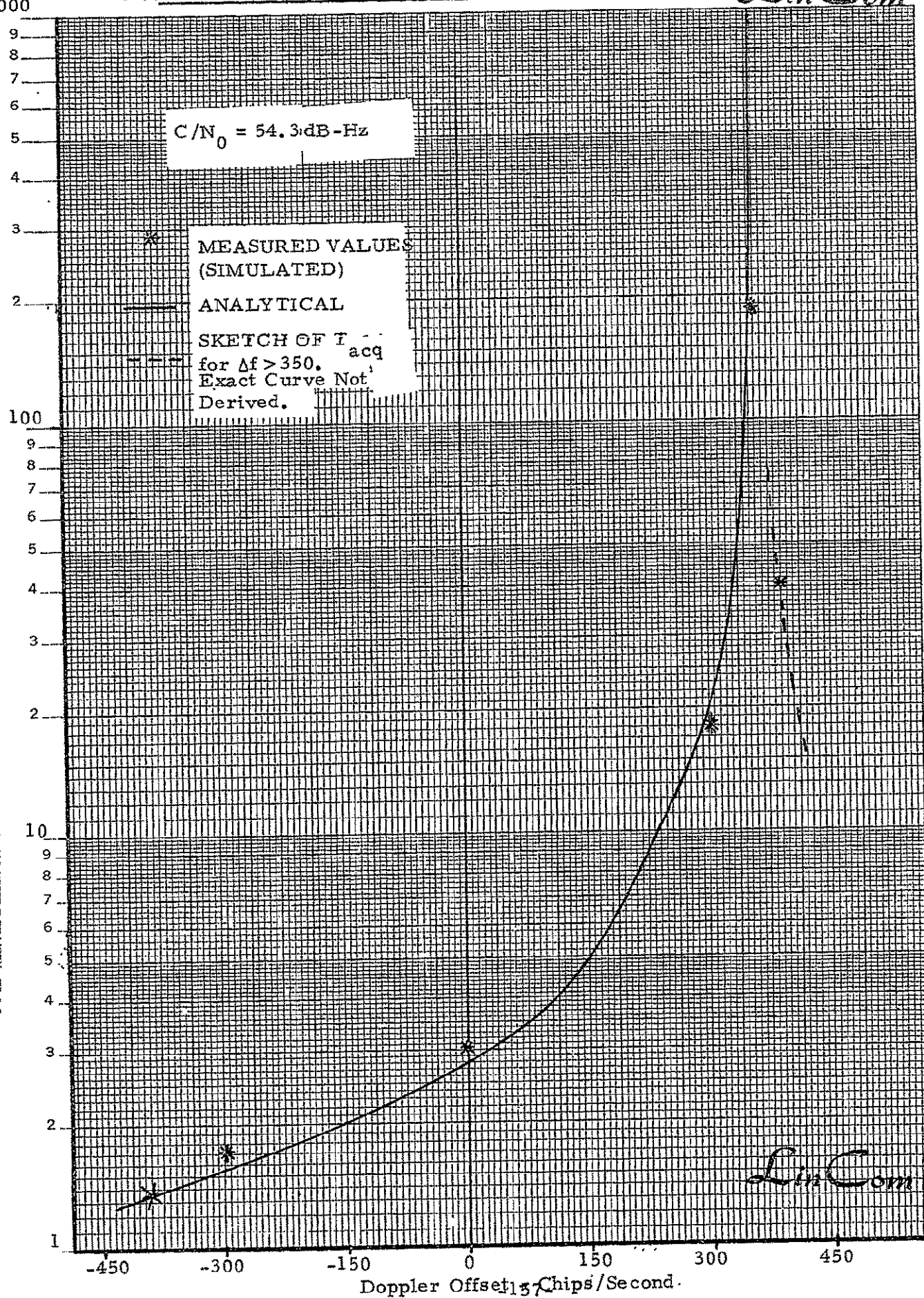
where Δf is the doppler offset and T_D is the mean dwell time at each of the 2047×2 search positions. The parameter T_D is a function of the basic search time per cell, search algorithm, and threshold parameter settings. A curve of T_{acq} versus Δf is plotted in Figure 6.19 and indicates that for the choice of algorithms, threshold settings, etc.,

FIGURE 6.19. ACQUISITION TIME VERSUS DOPPLER OFFSET.

LinCom

46 5813

K&E SEMI-LOGARITHMIC 3 CYCLES x 140 DIVISIONS
KEUFFEL & ESSER CO. MADE IN U.S.A.



LinCom

sync is not possible under positive doppler conditions around 350 code chips/second. As the doppler is increased above this value sync is once again possible.

6.6.2 Doppler Limitations Due to Loop Behavior

For a B_L of 200 Hz, the no noise pull in frequency of the τ -dither loop was experimentally measured to be $\approx \pm 440$ chips/second. Figure 6.20A depicts sketches of the reference phase, ϕ_R , and incoming phase, ϕ_i , near code alignment for positive and negative doppler value less than 440 chips/second. For this condition the loop always acquires. Figure 6.20B displays sketches for doppler greater than 440 chips/second. For a negative value the local code phase will step up to near alignment, but the loop will not lock. The next step of ϕ_R moves the code phases further apart. For positive doppler the local code phase will step up to near alignment, the loop once again does not pull in, but ϕ_R remain below ϕ_i . When the search commences ϕ_R will step up once again to ϕ_i , the loop tries to acquire but fails and ϕ_R remains below ϕ_i . In summary the system will continue to hunt about the sync point. The sketch shown assumes a dwell time doppler product greater than one chip which corresponds to our situation.

Possible phase trajectories for lower doppler conditions and noise present are sketched in Figure 6.21A. A situation is shown for positive doppler whereby the reference phase is stepped near alignment, sync is not detected or the loop does not track, with ϕ_R remaining below ϕ_i . Multiple passes within the same epoch become possible until PN acquisition and loop lock simultaneously occur. This hunting action for acceptable positive doppler values significantly improves performance; no improvement occurs for negative doppler.

The sketches of Figure 6.21B represent the condition where noise injected into the loop finally moves ϕ_R above ϕ_i for positive doppler, and

46 1513

10 X 10 TO THE CENTIMETER 12 X 25 CM
KEUFFEL & ESSER CO. MADE IN U.S.A.

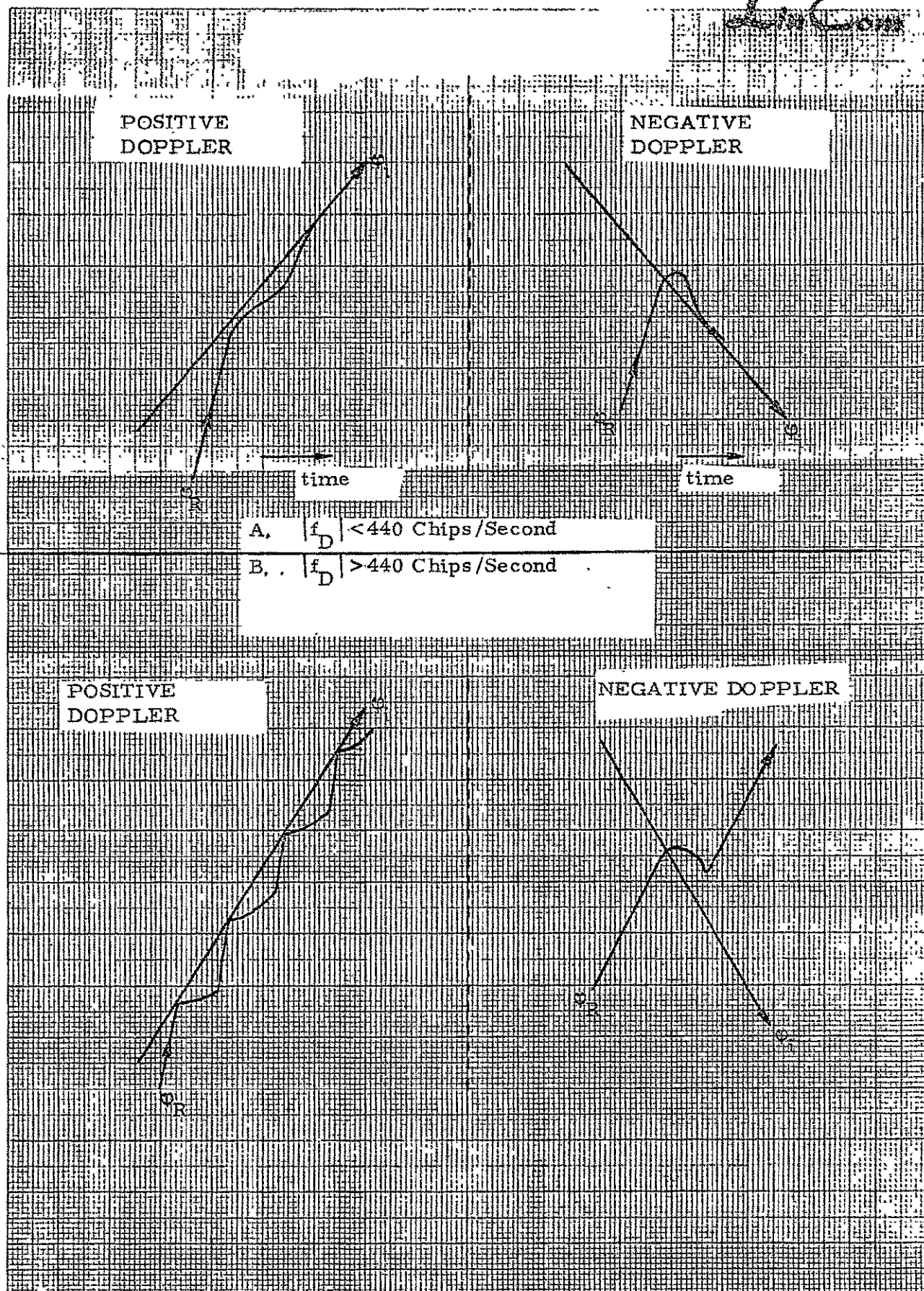


FIGURE 6.20.
CODE PHASE WAVEFORMS, NO NOISE.

LinCom

POSITIVE
DOPPLER

NEGATIVE
DOPPLER

A. $B_L = 200 \text{ Hz}$, $|f_D| = 300 \text{ Chips/Second}$

B. $B_L = 300 \text{ Hz}$, $|f_D| = 300 \text{ Chips/Second}$

FIGURE 6.21.
CODE PHASE WAVEFORMS, NOISE.

below ϕ_i for negative doppler. In this situation positive doppler acquisition performance is degraded while negative doppler performance is enhanced. A question one might ask at this point is can the situation sketched in Figure 6.21B occur? The answer as far as the simulation is concerned is negative since we always reset our phase reference, ϕ_R , to the value it had prior to searching a particular cell (assuming sync is not detected). Could it occur if the phase was not reset? Data from the results of an experiment evaluating loop behavior are tabulated in Table 6.4 and indicates that this condition is possible but not very likely for a loop bandwidth of 200 Hz but is very possible for TRW's B_L of 300 Hz.

6.6.3 Discussion

For the parameters and algorithms used in the simulation acquisition was first not possible at code doppler values around 350 code chips per second. This hang-up being independent of the delay lock loop. As doppler moved above this point acquisition once again became possible with the loop going out of lock more frequently, thus causing us to dwell near the sync point for longer periods of time before acquiring. For the loop bandwidth used and due to a strategy of resetting the code reference no noticeable degradation was observed related to phase jitter driving us away from the dwell point.

A direct answer to TRW's sync problem cannot be given. Data obtained from the simulation indicates that TRW's performance will be somewhat but not totally degraded due to their higher value of B_L and increased possibility of jitter driving them in the wrong direction. A total hang-up condition such as depicted in Figure 6.20B could however be caused by some hardware problem. In either of these cases reducing the positive doppler offset could reduce acquisition time and allow them to meet spec.

TABLE 6.4. LOOP BEHAVIOR EXPERIMENT

Doppler Offset Chips/Sec	Number of Trials Loop Walked Out of Sync Prior to Bandwidth Reduction	Number of Trials Loop Walked Out of Sync In Direction that would not allow the possibility for further code alignment
0*	6	3
+300*	6	0
+300 ⁺	8	4
+350*	6	0
+400*	9	1
+400 ⁺	10	6
Total Number of Trials = 15 $C/N_0 = 48.3 \text{ dB-Hz}$ $*B_L = 200 \text{ Hz} \quad +B_L = 300 \text{ Hz}$		

SECTION 7

CONCLUSIONS

The Phase II objectives were:

1. Further investigate the PN Subsystem acquisition behavior.
2. Support NASA/JSC personnel in their use of the hardware simulation.
3. Construct a Symbol Synchronizer Subsystem which closely emulates the expected hardware.
4. Integrate the Symbol Synchronizer Subsystem into the composite simulation of the receiver.
5. Perform a series of system studies to evaluate overall performance of the receiver when subjected to anomalies such as signal fades and detected potential problems associated with PN/Carrier sweep interactions.

The investigations associated with meeting these objectives have been documented in the preceding five sections. An individual section being, for the most part, completely dedicated to a single objective. Conclusions relating to each objective are presented within these sections.

Some of the key conclusions reached are summarized in the following paragraphs.

The curves of Figure 2.18 represent the simulated prediction of the composite PN Subsystem acquisition performance for the LinCom based sync algorithms. These curves take into account the effects of loop behavior, noise estimate, switching transients,

and filtering in their prediction of performance. At a C/N_0 of 48.3 dB-Hz the mean acquisition time is seen to be about 50 seconds and is relatively insensitive to doppler.

Figures 3.7 and 3.9 present evidence that the ESTL PN acquisition algorithms using up/down counters is superior to LinCom's use of reset counters when one considers loop behavior.

Preliminary comparison of the measured performance of the Harris despreader with that predicted by the simulation indicate that good agreement exists. It is expected that excellent agreement will be found once the software package is exercised by NASA in direct support of hardware testing in a carefully controlled and monitored environment.

At a design point loop bandwidth of 30 Hz there will be very little noticeable degradation in performance associated with the Symbol Synchronizer Subsystem. When this subsystem is integrated into the composite simulation, its contribution to the system degradation can be severe. The loss being a function of the 2nd IF filter bandwidth and sampling rate associated with the digitally implemented synchronizer. This degradation being about .7 dB for an IF bandwidth equal to four times the symbol rate and a sampling frequency equal to ten times the symbol rate. Loss being reducible by increasing the sampling rate or removing the IF filter or through channel equalization.

The study evaluating the system response to signal fades indicates that sync detector hardware and algorithms and not timing loops are limiting in establishing performance. Carrier phase discontinuities due to switching transients don't appear to pose a serious problem.

Experiments performed indicate that both the PN and Carrier Recovery Subsystems are relatively insensitive to carrier sweep, at least for sweep rates of interest.

Results from the system reconfiguration experiments indicate that (1) for the design point parameters used, tracking performance of the receiver is not sensitive to Costas arm filter variations; acquisition properties were not evaluated; (2) with the SSS present the optimum IF filter bandwidth is infinite but, if one is bandlimited, it appears that there is a "best" bandwidth setting which is a function of sampling rate.